



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**

DEPARTMENT OF INFORMATION SYSTEMS

**APLIKACE PRO ŘÍZENÍ AUDITŮ**

AUDIT MANAGEMENT APPLICATION

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**ROMAN VALENA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. RADEK BURGET, Ph.D.**

**BRNO 2017**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav informačních systémů

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Valena Roman**

Obor: Informační technologie

Téma: **Aplikace pro řízení auditů**  
**Audit Management Application**

Kategorie: Informační systémy

**Pokyny:**

1. Prostudujte současné technologie pro tvorbu aplikací s webovým uživatelským rozhraním.
2. Seznamte se s požadavky na systém pro řízení auditů na výrobních linkách.
3. Na základě analýzy požadavků navrhnete systém pro plánování auditů, evidenci jejich výsledků a sledování řešení odhalených problémů.
4. Implementujte navržený systém pomocí vhodně zvolených technologií.
5. Implementujte funkce pro grafickou prezentaci výsledků a export dat pro další zpracování.
6. Proveďte testování systému a zhodnoťte dosažené výsledky.

**Literatura:**

- Gutmans, A., Rethans, D., Bakken, S.: Mistrovství v PHP 5, Computer Press, 2012
- Žára, O.: JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

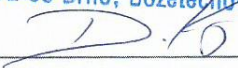
Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Burget Radek, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2

  
doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

## Abstrakt

Tato bakalářská práce popisuje vývoj webové aplikace pro řízení auditů ve firmě THK Rhythm Automotive Czech, která se zabývá výrobou automobilových součástek. Aplikace umožňuje efektivnější plánování a provádění auditů. Monitoruje odhalené problémy a způsob jejich řešení. Usnadňuje vytváření textových a grafických reportů. V úvodu je analyzována problematika původního systému řízení auditů. Na základě analýzy je vytvořen návrh aplikace. Obsažen je také popis samotné implementace, použité technologie a způsob integrace do firemního prostředí.

## Abstract

This bachelor's thesis describes the development of a web application for audit management in THK Rhythm Automotive Czech company, which is engaged in manufacturing of automobile parts. The application helps to plan and execute audits more effectively. It makes creating text and graphic reports much easier. The introduction analyzes the current way of audit management. Based on the analysis, design of the application is proposed. The description of the implementation, used technologies and the way of integration into the company interface is included too.

## Klíčová slova

Webové technologie, Informační systém, Řízení auditů, PHP, MS-SQL, jQuery, jqPlot, iCalendar

## Keywords

Web technologies, Information System, Audit management, PHP, MS-SQL, jQuery, jqPlot, iCalendar

## Citace

VALENA, Roman. *Aplikace pro řízení auditů*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Burget Radek.

# Aplikace pro řízení auditů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Radka Burgeta, Ph. D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Roman Valena  
18. května 2017

## Poděkování

Děkuji panu Ing. Radku Burgetovi, Ph.D. za odborný dohled, cenné rady, vedení mojí bakalářské práce a projevenou trřlivost. Děkuji také firmě THK Rhythm Automotive Czech, která mi umožnila vypracovat tuto bakalářskou práci.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
1.1	Společnost THK	3
1.2	Motivace	3
1.3	Cíl práce	3
<b>2</b>	<b>Specifikace požadavků</b>	<b>5</b>
2.1	Struktura výroby a zaměstnanců	5
2.2	Audit	6
2.2.1	Struktura formuláře	7
2.3	Proces auditu od naplánování až po uzavření	8
2.3.1	Plánování auditů	9
2.3.2	Provedení auditu	9
2.3.3	Reakce na výsledek	9
2.3.4	Reporty a grafy	10
2.4	Požadavky na novou aplikaci	10
<b>3</b>	<b>Použité technologie</b>	<b>11</b>
3.1	HTML	11
3.2	CSS	11
3.3	PHP	12
3.3.1	PHP Frameworky	12
3.4	JavaScript	12
3.4.1	jQuery	12
3.4.2	jQplot	12
3.5	Microsoft SQL Server	12
<b>4</b>	<b>Návrh aplikace</b>	<b>14</b>
4.1	Diagram případu užití	14
4.1.1	Guest	14
4.1.2	Řešitel	14
4.1.3	Auditor	14
4.1.4	Mistr	15
4.1.5	Hlavní auditor	15
4.1.6	Administrátor	15
4.1.7	Server	16
4.2	Návrh databáze	16
4.2.1	Entity a fyzický model databáze	17
4.3	Návrh uživatelského rozhraní	19

<b>5 Implementace</b>	<b>20</b>
5.1 Model životního cyklu aplikace . . . . .	20
5.2 Přihlášení do aplikace – integrace do současného systému . . . . .	21
5.3 Správa uživatelů . . . . .	22
5.4 Aktualizace vztahu buňka-mistr . . . . .	22
5.5 Aktualizace otázek a aktualizace typu auditu . . . . .	23
5.5.1 Otázka vyskytující se ve více formulářích . . . . .	23
5.6 Notifikace prostřednictvím emailu . . . . .	25
5.7 Automatické plánování událostí do kalendáře Microsoft Outlook Express . .	26
5.7.1 iCalendar . . . . .	26
5.8 Grafická prezentace výsledků . . . . .	29
5.9 Export dat do CSV . . . . .	31
5.10 Ošetření výjimečných případů . . . . .	32
<b>6 Testování a uvedení do provozu</b>	<b>33</b>
6.1 Testování aplikace v průběhu vývoje . . . . .	33
6.2 Zaškolení uživatelů . . . . .	34
6.3 Zkušební provoz . . . . .	34
<b>7 Závěr</b>	<b>35</b>
<b>Literatura</b>	<b>37</b>
<b>Přílohy</b>	<b>38</b>
<b>A Obsah CD</b>	<b>39</b>
<b>B Rozdělení uživatelského menu</b>	<b>40</b>
<b>C Ukázka aplikace</b>	<b>41</b>

# Kapitola 1

## Úvod

### 1.1 Společnost THK

Společnost THK<sup>1</sup> se zabývá vývojem mechanismů lineárního pohybu. Historie této firmy sahá až do roku 1971. Od této doby rozšířila své pole působnosti na tři světové kontinenty – Amerika, Asie a Evropa. Své zastoupení má i v České republice. Konkrétně se jedná o výrobní závod THK Rhythm Automotive Czech ( “*dále TRA* ”) se sídlem v jižních Čechách v obci Dačice. V TRA se specializují především na výrobu automobilových dílů. Mezi zákazníky patří věhlasné světové automobilky, jako například Audi, BMW, VW, Mercedes-Benz, Jaguar, Volvo, Porsche, Ferrari a mnoho dalších.

### 1.2 Motivace

Pokud si chce firma udržet svou konkurenceschopnost, musí neustále vyvíjet a zlepšovat své pracovní postupy a technologie. Požadavky zákazníků jsou čím dál náročnější. Důraz je kladen především na kvalitu, cenu a čas, za který je firma schopna díly vyrobit a připravit k expedici. Do optimalizace těchto oblastí je investováno velké množství úsilí, peněz a času. Každá součástka má přesně daný výrobní postup. Pro dosažení optimálních výsledků je nutné tyto postupy správně dodržovat. Bohužel ani stroje, ani zaměstnanci, kteří na těchto strojích pracují, nejsou neomylní. Důsledkem toho vznikají nežádané situace, které je potřeba v co nejkratším čase odstranit a vyvodit patřičné důsledky. Jedním ze způsobů, jak kontrolovat správné dodržování pracovních a výrobních postupů a evidovat vzniklé problémy, je provádění auditů.

Audit má za úkol zjistit aktuální stav na daném pracovišti a odhalit případné nedostatky. Pokud je na pracovišti zjištěn problém, je potřeba informovat osobu, která za danou oblast odpovídá. Jejím úkolem je analyzovat danou situaci a odstranit všechny nedostatky. Na základě analýzy výsledků z provedených auditů je možné lépe cílit na nejkritičtější místa a efektivně tak předcházet vzniku těchto problémů.

### 1.3 Cíl práce

Současná forma celého procesu provádění auditů není příliš efektivní a všechny operace jsou řešeny ručně. Výsledek prováděné kontroly je zapsán do papírového formuláře. Tento

---

<sup>1</sup><http://www.thk.com>

záznam je pak naskenován do společné složky pro další zpracování. Informace musí být poté přepsány do textové podoby. Ne všechny formuláře jsou neprodleně po provedení auditu zpracovány. Evidované informace tak nejsou vždy aktuální. Často není známo, zda byl audit již proveden a jaký byl jeho výsledek. U nevyhovujících auditů není známo, zda se na odstranění problému stále pracuje a zda se začalo na odstranění problém vůbec pracovat. Osoby, neplnící si své povinnosti musí být opakovaně upomínáni.

Zpracování získaných poznatků, jejich vyhodnocování a následná prezentace probíhá v programu Microsoft Excel. Data jsou prezentována textově i graficky. Nevýhodou je statická podoba těchto informací – jsou pevně dány bez možnosti vyfiltrovat si data podle konkrétních kritérií.

Všechny evidované informace jsou uloženy v několika složkách na firemním serveru společně s mnoha dalšími daty. Pro neznalou osobu je velmi těžké se v takovém množství dat orientovat. Některá informace jsou součástí více souborů. Mezi těmito soubory není přímá vazba. Při změně takových informací je potřeba tuto změnu zanést do všech ostatních souborů. Vlivem nepozornosti tak mohou vznikat nekonzistentní data. Celý systém je velmi chaotický.

Cílem této práce je vytvořit aplikaci, která pomůže odstranit tyto nedostatky a tím zrychlit, zpřehlednit a do jisté míry zautomatizovat celý proces auditů.



## Kapitola 2

# Specifikace požadavků

Analýza je často podceňována, avšak jedná se o jednu z nejdůležitějších částí celého vývoje. Je potřeba důkladně zjistit všechny požadavky zákazníka a ujistit se, že daný problém chápou obě strany stejným způsobem.

V první části si lehce přiblížíme, jak funguje systém výroby automobilových součástek. Ukážeme si fyzické rozdělení výrobní haly na jednotlivé úseky a strukturu zaměstnanců a jejich role.

Jakmile budeme seznámeni s první částí, můžeme přistoupit k objasnění toho, co to audit vůbec je a jaké jsou jeho vlastnosti. Postupně probereme jednotlivé fáze provádění kontroly od začátku až dokonce.

Aplikaci je potřeba integrovat do již zaběhlého systému firmy. Výčet těchto podmínek je popsán v poslední části této kapitoly.

### 2.1 Struktura výroby a zaměstnanců

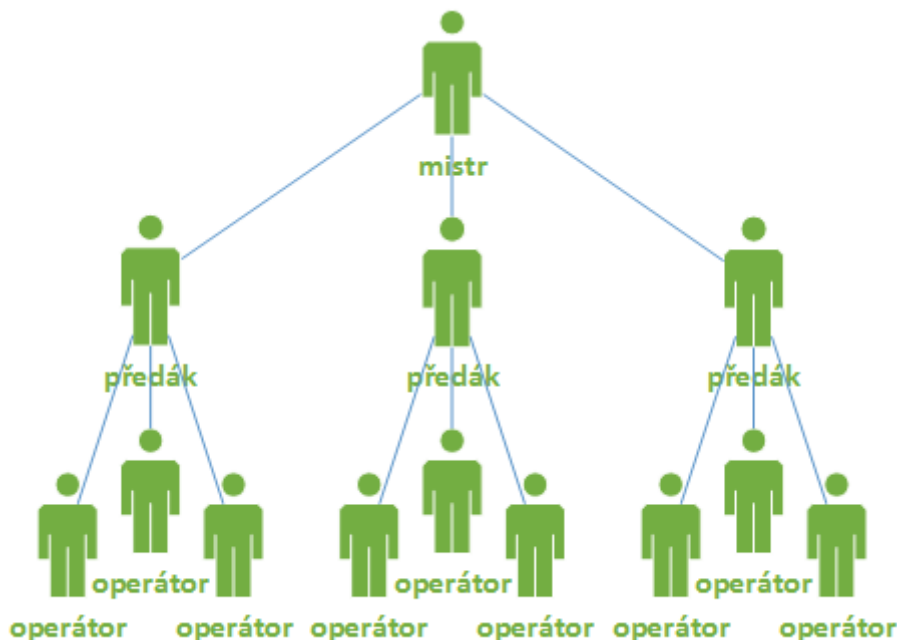
Nejprve se podíváme na výrobní proces z fyzického pohledu a poté si ukážeme vztahy mezi jednotlivými zaměstnanci dílny.

Než je součástka připravena k expedici, projde během svého životního cyklu posloupností několika operací. Každá operace je prováděna na k tomu předepsaných stanovištích – buňkách. Jedná se o výrobní linku, která je tvořena jedním nebo více stroji, které jsou seřizeny na konkrétní operaci a konkrétní díl. Soubor několika buněk pak vytváří **středisko**. Struktura výrobní haly je na obrázku 2.1.



Obrázek 2.1: Způsob rozdělení výrobní haly

V každé firmě by mělo být jasné, jakou roli zastává každý zaměstnanec a co se od něj očekává. Neméně důležitou informací je pravomoc a vztah k ostatním zaměstnancům. K popisu těchto informací se využívají organizační struktury. Pro účely této práce nám postačí znát pouze strukturu zaměstnanců určenou pro oblast výroby. Vztahy mezi zaměstnanci vyššího managementu nejsou pro pochopení zkoumané problematiky důležité.



Obrázek 2.2: Struktura zaměstnanců

Z obrázku 2.2 je patrné, že nejvyšší pravomoc má **mistr**. S rozsahem pravomoci úměrně stoupá i míra odpovědnosti. Každý mistr má na starost právě jedno středisko. Mistr komunikuje s vyšším managementem a podle požadavků organizuje výrobu mezi jednotlivé buňky a svůj tým. Je odpovědný za výkony a bezproblémový chod celého střediska. V případě vzniku nepředvídatelné události musí na tuto událost patřičně reagovat.

Ve firmě se pracuje na tři směny – ranní, odpolední a noční. Mistr nemůže být přítomen na každé směně. V době jeho nepřítomnosti zastupuje jeho roli **předák**. Na konci nebo na začátku každé směny podává předák hlášení o průběhu celé směny mistrovi. Další náplní jeho práce je seřizování výrobních strojů a školení nových zaměstnanců.

Na nejnižší úrovni se nachází **operátor**. Každému operátorovi je před začátkem směny přiřazena buňka, na které odvádí svůj výkon. Je odpovědný za své výkony, stav svého pracoviště a dodržování pracovních postupů. V případě problémů se obrací na předáka.

## 2.2 Audit

Každý díl má specifický výrobní postup, každý stroj doporučené parametry a každý operátor zásady, které musí být dodržovány. Proto se na pracovištích provádějí kontroly, které se zkoumají, zda dochází k plnění těchto klíčových ukazatelů. Provedení auditu znamená vyplnění předem připraveného formuláře.

Podle oblasti, na kterou se kontrola zaměřuje, rozlišujeme dva typy auditů, konkrétně LPA audit<sup>1</sup> a HSE audit. Princip provádění auditů je téměř stejný, proto se zatím spokojíme s informací, že existují typy dva. Konkrétní odlišnosti budou podrobněji popsány v následujících řádcích.

### 2.2.1 Struktura formuláře

Formulář, který je potřeba při provedení auditu vyplnit, je složen z hlavičky a těla formuláře. Hlavička a část těla je zobrazena na obrázku 2.3.

#### HLAVIČKA

Společné informace pro oba typy auditů:


- *datum provedení*
- *auditor*, který audit provedl
- *typ auditu*
- *buňka*, které se audit týkal
- *mistr*, odpovědný za buňku
- *plánované datum provedení*
- *skutečné datum provedení*

Pouze LPA:

- *díl*, který se na lince vyrábí (pouze LPA audit)

Pouze HSE:

- *jmeno předáka*, který byl na směně přítomen
- *jmeno hodnoceného* – operátor, který obsluhoval buňku

	<b>Layered Process Audit</b>						DATUM :                      Mistr:		
							LINKA :                      Díl:		
	Frekvence auditu			M		T		D	AUDITOR :
Otázky	č.	Návod	OK	NOK	Okamžitá opatření			Komentář	
Identifikace a pořádek	1	Nejsou zde kusy bez identifikace (zbytková množství, jiné díly)?			Žádat operátora, aby dal kusy bez identifikace nebo podezřelé kusy okamžitě mezi zmetky				
Kontroly a záznamy	2	Ujistěte se, že jsou veškeré kontroly prováděny správným způsobem. Zeptejte se operátorů, co za kontroly mají provádět a s jakou frekvencí. (musí znát 100% kontrolu I. četnost kalibrace měřidel za směnu atd.) Zkontrolujte, je-li zajištěno provádění TESTU a SPC dle ICP.			Zajistit okamžitě proškolení předákem.				
	3	Zkontrolujte, jestli jsou kontroly pravidelné a správně zaznamenávány. Požádejte operátora o zapsání jednoho kusu před vámi. (SPC, záznam o kontrole 1. kusu...)			Zajistit okamžitě proškolení předákem.				

Obrázek 2.3: Ukázka rozložení LPA formuláře

#### TĚLO

Tělo je složeno z několika otázek, na které musí auditor odpovědět. Otázka odpovídá jednomu sledovanému aspektu konkrétního pracoviště a má tyto vlastnosti:

<sup>1</sup>Layered process audit

Společné pro všechny audity:

- *kategorie* – která je předem dána
- *text otázky* – kritéria, která musí dané pracoviště splňovat
- *odpověď* – vyplní auditor, na výběr jsou tři možnosti
  - OK – aktuální stav splňuje požadovaná kritéria
  - NOK – aktuální stav nesplňuje požadovaná kritéria otázky
  - n/a – otázku nelze hodnotit
- *celkový komentář* – volitelná položka
- *výsledek* – počet OK otázek z celkového počtu hodnocených otázek

Pouze HSE:

- *komentář k otázce* – v případě NOK otázky<sup>2</sup> je nutné vysvětlit důvod NOK hodnocení

Pouze LPA:

- *okamžité opatření* – každá otázka typu LPA má předem stanovený návod jak postupovat při řešení NOK otázky
- *typ závady* – v případě NOK otázky je nutné popsat závadu
- *komentář k otázce* – volitelný komentář

Pro lepší ilustraci příklad jedné vyhodnocené LPA otázky:

*Kategorie:* Kontroly a záznamy

*Text otázky:* Zkontrolujte, jestli jsou kontroly pravidelně a správně zaznamenávány. Požádejte operátora o zapsání jednoho kusu před vámi.

*Okamžité opatření:* Zajistit okamžité proškolení předákem.

*Odpověď:* NOK

*Typ závady:* operátor nezná postup

## 2.3 Proces auditu od naplánování až po uzavření

Celý proces se dá chápat jako posloupnost několika akcí.

1. **Naplánování kontroly** na určité datum
2. **Provedení kontroly** – vyplnění předpřipraveného formuláře
3. **Reakce na výsledek** – v případě zjištění chyby při prováděné kontrole je potřeba tuto chybu odstranit
4. **Evidence celého procesu** – zpracování dat z provedeného auditu a případné reakce

V celém procesu se setkáme celkem se čtyřmi typy funkcí.

**Hlavní auditor** – plánuje audity, dohlíží na včasný a korektní průběh auditů, vytváří re-

---

<sup>2</sup>Jako NOK otázka je označena taková otázka, která nevyhovuje požadovanému kritériu

porty

**Auditor** – provádí audit

**Mistr** – odpovídá za způsob, jakým je vyřešen problémový audit

**Řešitel** – jeho úkolem je opravit závadu, která byla zjištěna při provádění kontroly

### 2.3.1 Plánování auditů

#### Plánované audity

Na konci každého roku vytváří hlavní auditor časový harmonogram pro rok nadcházející. Každý audit musí mít jasně přidělenou buňku a plánované datum provedení. U LPA auditu je navíc předem znám i auditor. U HSE auditu je výběr auditora v kompetenci mistra. Podmínkou pro korektní plán je minimálně jedna kontrola za rok na každé buňce.

#### Neplánované audity

Vykonáváním neplánovaných auditů jsou pověřeni speciální auditoři. Kontroly provádí na denní bázi a plánují si obchůzky spontánně v průběhu dne.

### 2.3.2 Provedení auditu

Úkolem auditora je provést kontrolu v týdnu, na který byl audit naplánován. Svědomitě vyplněný formulář je nutné vzápětí naskenovat a umístit do příslušné složky na server. O výsledku auditu by měl být informován mistr nezávisle na tom, jak výsledek dopadl. V případě, že nebyl audit vyhotoven v týdenním časovém horizontu, je klasifikován a evidován jako “neprovedený včas”.

### 2.3.3 Reakce na výsledek

V počáteční fázi je audit ve stavu:

- **NAPLÁNOVANÝ** – nebyl proveden

Po provedení se může nacházet v jednom z následujících tří stavů:

- **PROVEDENÝ OK** – audit byl proveden v pořádku
- **ŘEŠÍ SE** – audit obsahoval jednu nebo více problémových otázek, na odstranění chyby se pracuje
- **PROVEDENÝ NOK** – audit obsahoval jednu nebo více problémových otázek a všechny chyby byly odstraněny

Mistr reaguje na audit pouze při zjištění jednoho či více nedostatků. Podle charakteru závady se rozhodne pro nejefektivnější způsob řešení. Nápravu může provést sám a nebo tento problém delegovat na další osobu ze svého týmu. Jeden audit může řešit více řešitelů. Jakmile jsou vyřešeny všechny komplikace, audit přechází do stavu **PROVEDENÝ NOK** a je uzavřen.

### 2.3.4 Reporty a grafy

Na základě evidovaných dat jsou vytvářeny reporty a statistiky. Informace jsou interpretovány jak textově, tak i formou grafického zobrazení. Všechny operace jsou prováděny ručně v programu Microsoft Excel. Množina zkoumaných oblastí je omezena na ty nejdůležitější statistiky.

- počet provedených auditů / počet plánovaných auditů
- počet provedených NOK auditů <sup>3</sup> / počet provedených auditů
- počet vyřešených NOK auditů / počet provedených auditů
- počet NOK otázek
- nejproblémovější otázky

## 2.4 Požadavky na novou aplikaci

Z výše popsané specifikace lze intuitivně vyvodit některé požadavky. Filozofie fungování aplikace by měla být stejná, jako je neaplikační provádění auditů, avšak využití webových technologií nám přináší nové možnosti.

Souhrn požadavků na aplikaci:

- Administrace uživatelů, kategorií otázek, textu otázek, typů auditu
- Ruční plánování auditů
- Automatické plánování auditů do kalendáře Outlooku Express
- Automatické upozornění na provedený audit
- Automatické opakované upozornění na nevyřešený audit
- Prohlížení, provedení, řešení, editace auditu
- Zobrazení statistik podle zvolených kritérií
- Grafická prezentace výsledků
- Export dat do souboru – formát CSV (viz kapitola 5.4.)

---

<sup>3</sup>NOK audit je audit, který obsahuje jednu nebo více NOK otázek

## Kapitola 3

# Použité technologie

Při výběru správné technologie pro implementaci budeme vycházet z požadavků na funkcionalitu aplikace. Musíme také zohlednit softwarové možnosti zadavatele.

Aplikace bude pracovat s velkým množstvím dat, které budeme ukládat do databáze. Mezi nejrozšířenější databázové servery patří MySQL. My využijme služby SQL Serveru od firmy Microsoft, který je ve firmě TRA standardně používán.

Uživatelské rozhraní pro komunikaci uživatele s aplikací vytvoříme stejným způsobem, jako je tvořena většina podobných aplikací v TRA – jako webovou aplikaci. Webové stránky vytvoříme za pomoci standardních jazyků k tomu určených. Pohodlnější a komfortnější práci s aplikací nám pomůže zajistit JavaScript a její knihovny, jako například jQuery. Komunikaci mezi webovým prohlížečem a databází nám zprostředkuje programovací jazyk PHP. Aktuální verze php, která běží na serverech firmy TRA, je verze php 5.2. Aplikaci je potřeba optimalizovat tak, aby byla plně kompatibilní s webovým prohlížečem Internet Explorer(verze 11 a vyšší) a rozlišením 1280x1024 a vyšším.

### 3.1 HTML

*Hypertext Markup Language* je značkovací jazyk určený pro vytváření webových stránek. Jednotlivé informace jsou uzavřeny do tagů, které nám definují vlastnost dané informace. Pro interpretaci HTML dokumentu slouží programy – webové prohlížeče. Ne každý prohlížeč zpracovává dokumenty stejným způsobem, proto je vždy nutné stránky optimalizovat. pro každý prohlížeč zvlášť. Tak, jako každý jazyk, i HTML se stále vyvíjí. Dříve se pomocí HTML definoval i výsledný vzhled stránky. Zdrojový kód byl ale velice nepřehledný a těžce modifikovatelný. Proto se nyní používá pouze pro definici obsahu dokumentu. Pro účely popisu vzhledu dokumentu byl vytvořen jazyk CSS. [9]

### 3.2 CSS

*Cascading Style Sheets* – Kaskádové styly definují vzhled HTML dokumentů. Popisují, jak se bude výsledný obsah zobrazovat na displeji, papíru nebo na jiném médiu. Výhodou kaskádových stylů je to, že změna jediného elementu v css souboru se promítne do celého obsahu HTML dokumentu. To nám ušetří mnoho času. [8]

## 3.3 PHP

*Hypertext Preprocessor* – jedná se o jeden z nejpoužívanějších skriptovacích jazyků, který v roce 1994 vytvořil Rasmus Lerdorf. Předně je určen k tvorbě dynamických webových stránek a je vykonáván na straně serveru. Syntaxe jazyka je velmi podobná jazyku C. Od verze 3 podporuje také OOP. [1]

### 3.3.1 PHP Frameworky

Tak jako mnoho jiných jazyků existuje pro PHP celá škála různých frameworků. Každý má trochu jiné principy a naučit se s ním pracovat vyžaduje čas. Pokud si ale člověk osvojí základní praktiky, pozvedne svoje programování na vyšší úroveň a výrazně zvýší znovupoužitelnost kódu. Mezi celosvětově rozšířené patří například Laravel<sup>1</sup>, Symfony<sup>2</sup> nebo třeba Zend<sup>3</sup>. V České republice se ve značné míře užívá Nette<sup>4</sup>.

V TRA se pro vývoj webových aplikací nevyužívá žádný php framework. Požadavek byl implementovat aplikaci v čistém PHP, proto nebudeme uvedené frameworky rozebírat podrobněji.

## 3.4 JavaScript

JavaScript je objektově orientovaný skriptovací jazyk, který je vykonáván nejčastěji na straně uživatele. Lze ho ale použít i na straně serveru například v podobě Node.js. Používá se zejména pro tvorbu dynamických webových stránek. Pomocí JavaScriptu můžeme programovat různá interaktivní tlačítka a celkově tak zpříjemnit výsledný dojem a práci se stránkou.

### 3.4.1 jQuery

Jedná se o rychlou a funkčně bohatou JavaScriptovou knihovnu. Usnadňuje dynamickou změnu obsahu stránky, vytváření různých animací, práci s AJAXem. jQuery podporuje většina webových prohlížečů. <sup>5</sup>

### 3.4.2 jqplot

jqplot je zásuvný modul jQuery pro jednoduché vytváření interaktivních grafů v JavaScriptu. Způsob, jakým lze s tímto rozšířením pracovat je popsán blíže v kapitole 5.8. Mezi výhody patří jednoduchá instalace, přehledná dokumentace a široké množství funkcí. <sup>6</sup>

## 3.5 Microsoft SQL Server

MS-SQL server je relační databázový systém od firmy Microsoft. Existuje mnoho dalších databázových systémů jako například Oracle, nebo MySQL. Rozdíly mezi nimi jsou například v odlišné syntaxi některých SQL příkazů, množině podporovaných funkcí, robustnosti

---

<sup>1</sup><http://laravel.com/>

<sup>2</sup><http://symfony.com/>

<sup>3</sup><http://framework.zend.com/>

<sup>4</sup><http://nette.org/cs/>

<sup>5</sup><http://jquery.com/>

<sup>6</sup><http://www.jqplot.com/>



a celkové ceně. Pro uživatelsky pohodlnou správu MS-SQL databáze je využíván program Microsoft Managment Studio.

## Kapitola 4

# Návrh aplikace

Pomocí jazyka UML<sup>1</sup> budeme modelovat diagram případu užití jednotlivých uživatelů a navrhujeme model databáze. V posledním kroku se budeme zabývat uživatelským rozhráním, které je implementováno jako webové stránky. Zaměříme se na výsledný vzhled aplikace a způsob jejího rozdělení do funkčních celků.

### 4.1 Diagram případu užití

*Diagram případu užití* definuje jednotlivé role a jejich množinu případu užití. Jeden případ užití je chápán jako funkce, kterou systém vykonává jménem jednotlivých účastníků nebo v jejich prospěch. [4]

Rozlišovat budeme celkem šest účastníků – *administrátor*, *hlavní auditor*, *auditor*, *mistr*, *řešitel*, *guest*. Postupně si popíšeme každou roli zvlášť vzestupně podle jejich pravomocí. Tedy od nejmenší po největší.

#### 4.1.1 Guest

Pro práci s aplikací není přímo vyžadováno přihlášení. Takový uživatel má ale značně omezené možnosti – má práva pouze pro čtení. V aplikaci nemůže provádět žádné změny a ovlivňovat data v databázi. Může si prohlížet detaily auditů, jejich výsledky a statistiky.

#### 4.1.2 Řešitel

Každý uživatel, který je pověřen vyřešením alespoň jednoho auditu se automaticky stává řešitelem. V každém takovém auditu musí mít možnost popsat způsob odstranění problému. V případě auditu s větším počtem problémových otázek nemusí řešit všechny najednou. Každý problém může řešit zvlášť v delším časovém horizontu.

#### 4.1.3 Auditor

Auditoři se dělí celkem na tři úrovně, které jsou určeny podle jejich rolí ve firmě. Úroveň auditora nám také definuje oprávnění pro provádění auditu odpovídající úrovni:

- První úroveň – *LPA1 auditori*

---

<sup>1</sup>Unified Modeling Language – grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů

- Druhá úroveň – *technologové a mistři*
- Třetí úroveň – *Managment*

**Auditor první úrovně** má možnost plánovat LPA audity pro svoji potřebu (*tn. že ne-může plánovat audity nikomu jinému, než sobě*). Takové audity označujeme jako “neplánované LPA audity”.

**Auditoři druhé úrovně** jsou vázáni pouze předem daným plánem auditů pro celý rok a nemohou plánovat žádné nové audity.

**Managment** může, na rozdíl od první úrovně, plánovat pouze HSE audity – také pouze pro svoji potřebu.

Nezávisle na úrovni patří k dalším možnostem auditora provedení auditu, který mu přísluší. Při provádění auditu lze vyplňovaný formulář uložit takzvaně jako *draft* a k jeho dokončení se vrátit později.

#### 4.1.4 Mistr

Úkolem mistra je spravovat audity na svých buňkách. Termín auditů je vždy jasně daný. U LPA auditů je daný i auditor. U HSE auditů je při plánování nastaven jako auditor sám mistr, kterému auditovaná buňka náleží. Audit může provést sám, ale zpravidla ho vykonává jiná osoba z jeho týmu, kterou musí předem nastavit.

Dalším jeho úkolem je postarat se o řešení NOK auditů. Závady může vyřešit sám, nebo tímto úkolem pověřit jednoho nebo více uživatelů. V případě, že je audit vyřešen jinou osobou, než-li je sám mistr, musí mistr toto řešení zkontrolovat a potvrdit.

#### 4.1.5 Hlavní auditor

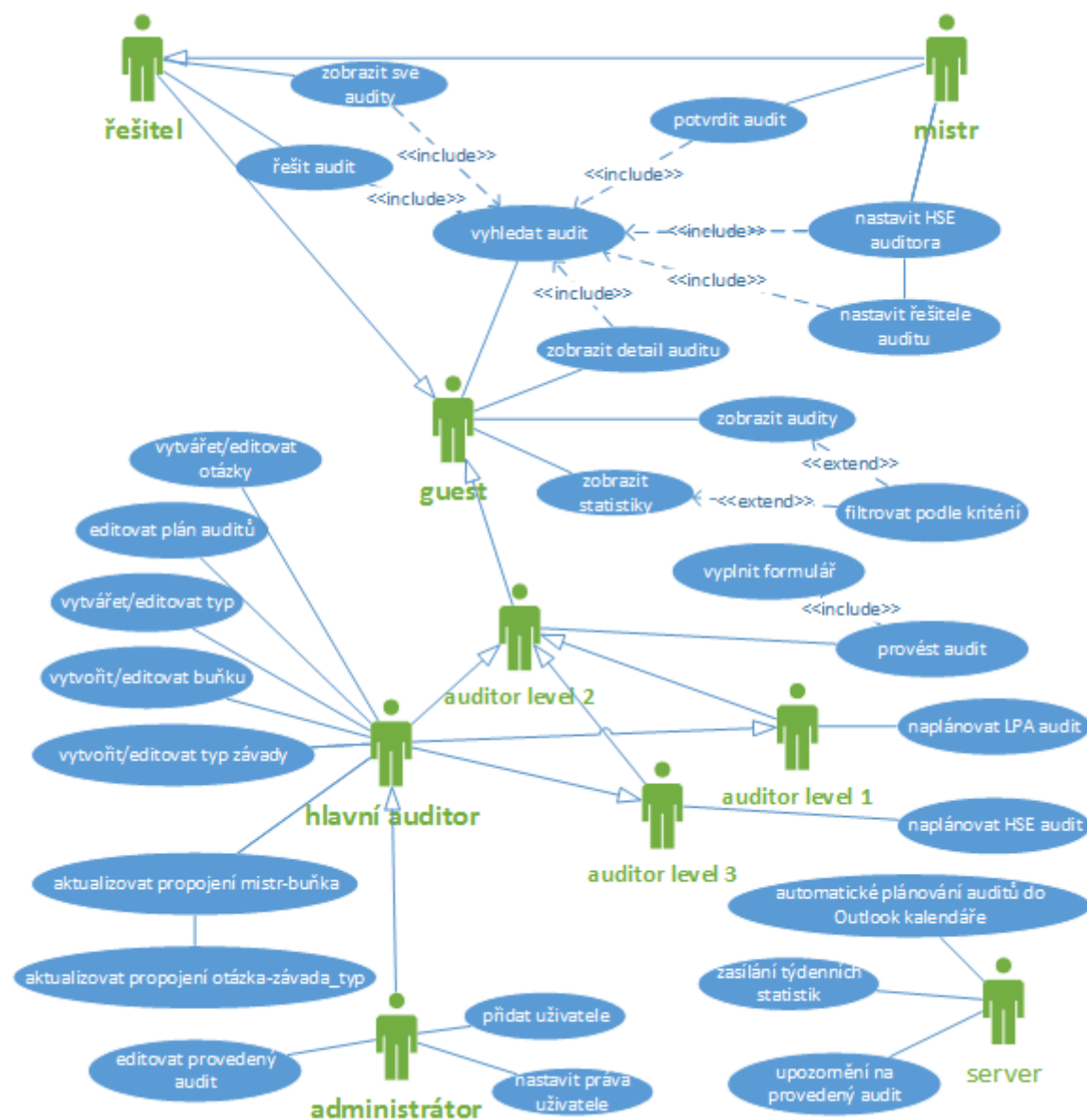
Hlavní auditor je manager všech auditů. Vytváří a edituje formuláře. Stará se o jejich aktuálnost a o aktuálnost jednotlivých otázek. U LPA otázek je při vyplňování formuláře nutné u NOK odpovědi uvést typ závady. Na každou otázku se vztahuje jiná množina možných závad. Tento seznam je potřeba v administraci definovat a udržovat. Stejně tak je potřeba udržovat správnou konexi mezi mistry a buňkami. Plánování auditů patří mezi další funkce hlavního auditora. Při vytváření plánu nastavuje každé kontrole typ auditu, auditora, buňku a datum provedení. Tyto informace může změnit v případě, že audit nebyl ještě proveden. Stejná podmínka musí být dodržena při odstranění naplánovaného auditu. Vyhodnocování statistik a upozorňování na neprovedené audity bude v již aplikaci prováděno automaticky.

#### 4.1.6 Administrátor

Administrátor by měl mít kontrolu téměř nad celou aplikací. Přiděluje práva všem uživatelům a může upravovat většinu dat celé databáze. Některé audity jsou vyhodnoceny špatným způsobem. Může se jednat o chybně posouzená kritéria nebo jen o překlep, který je potřeba upravit. Proto má administrátor možnost editovat provedené audity i způsob jejich řešení. Tato funkce by měla být používána jen ve výjimečných případech po konzultaci s hlavními auditory. V žádném případě nesmí docházet ke zneužívání této funkce. Výsledky by byly zkreslené a celý systém provádění auditů by postrádal smysl. Ze stejného důvodu nesmí administrátor smazat vyhotovený audit. Pokud tak chce učinit, musí kontaktovat správce aplikace, který provede zákrok přímo v databázi.

### 4.1.7 Server

Server bude zasílat notifikační emaily, které jsou podmíněny vykonáním určité akce – provedením auditu. V určitém časovém intervalu budou každému uživateli zasílány emaily s upozorněním na neřešené audity. Server také zajistí plánování auditů do kalendáře Microsoft Outlooku Express.



Obrázek 4.1: Diagram případu užití

## 4.2 Návrh databáze

Dobrý návrh relační databáze je základem pro implementaci dobré aplikace. Kvalitní návrh nám pomůže vytvořit stabilní, rychlou a přehlednou databázi, která nebude obsahovat nekonzistentní data. Špatný návrh může naopak způsobit velké problémy, především u více

rozsáhlých databázových systémů. Při návrhu musíme počítat i s tím, že se aplikace bude postupně vyvíjet a rozšiřovat. Důležité je dodržovat základní pravidla UML modelování. Jedním z nich je rozložení všech prvků reálné situace na co nejmenší části – entity.

*Entita* – „věc“ nebo objekt reálného světa rozlišitelný od jiných objektů, o níž chceme mít informace v DB.

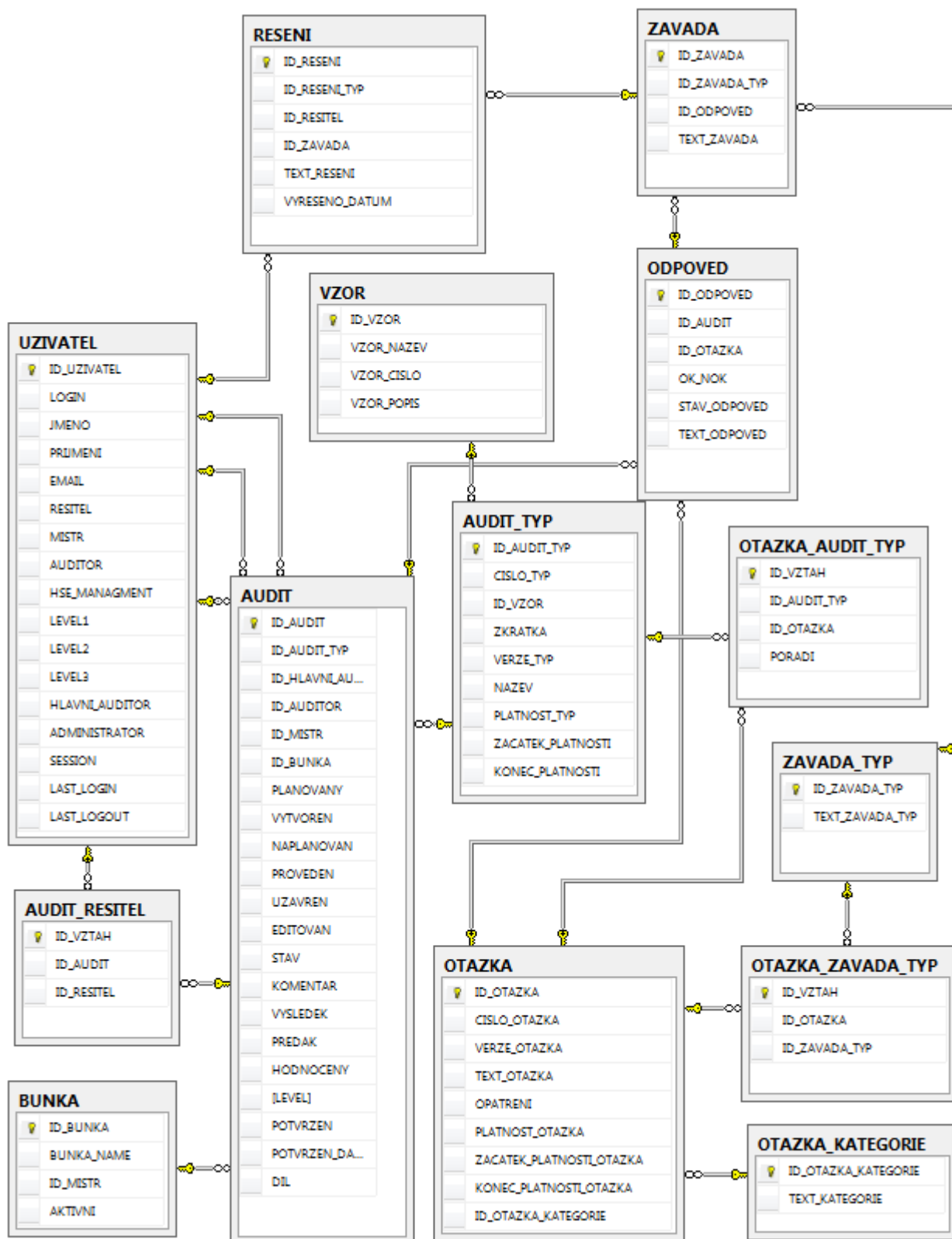
*Entitní množina* – množina entit téhož typu, které sdílí tytéž vlastnosti neboli atributy. [10]

#### 4.2.1 Entity a fyzický model databáze

Na základě požadavků, uvedených v druhé kapitole, byl návrh rozdělen celkem na tyto entitní množiny:

*uživatel, buňka, audit, typ auditu, vzor formuláře, otázka, kategorie otázky, odpověď na otázku, závada, typ závady, řešení závady*

Mezi entitami vytvoříme odpovídající relace a každé entitě přidělíme vlastnosti – atributy. Vznikne nám ER diagram, který poté převedeme na fyzický návrh databáze. Každá entitní množina je v databázi reprezentována tabulkou. Při vytváření tabulek a jejich atributů definujeme u každého atributu jeho datový typ, zda je povinný při vytváření nového záznamu a další možnosti, jako je označení primárních a unikátních klíčů. Výsledný fyzický model databáze je na obrázku 4.2 na straně 18.



Obrázek 4.2: Fyzický model databáze aplikace.

### 4.3 Návrh uživatelského rozhraní

Největší důraz aplikace je kladen především na její funkcionalitu. I přes to, že design aplikace není největší prioritou, je potřeba vytvořit přehledné a uživatelsky přívětivé prostředí, jehož ovládání bude intuitivní a logicky členěné.

Podle uživatelů, které jsme si definovali v kapitole 4.1, můžeme rozdělit uživatelské rozhraní na odpovídající funkční celky. Z administrace bude řízena celá aplikace. Plánování bude tvořit druhý funkční celek. Pro reportování výsledků bude vyčleněna další sekce. Hlavní jádro aplikace bude tvořit část auditů, rozdělena na “aktuální auditů”, “moje auditů” a “všechny auditů”.

Každý uživatel by měl mít ihned po přihlášení dokonalý přehled, kterým auditům je potřeba věnovat speciální pozornost a které nejsou v danou chvíli aktuální. Pro tyto účely je vytvořena stránka “aktuální auditů”. Na této stránce vidí uživatel všechny akce, které by měl neprodleně provést. Dělí se na celkem pět kategorií podle role uživatele – nastavení auditora, provedení auditu, vyřešení auditu, potvrzení auditu. V ideálním případě by měla být tato stránka prázdná. Respektive by měla uživatele informovat o skutečnosti, že jsou splněny všechny jeho povinnosti. Stránka “moje auditů” vypíše seznam všech auditů uživatele. Seznam je rozdělen podle role, kterou uživatel v daném auditu zastává. Kompletní seznam všech auditů v databázi je vypsán v poslední sekci “všechny auditů”. Pro jednoduché vyhledávání je možné využít vyhledávací filtrování.

Po logickém rozčlenění aplikace definujeme layout stránek. V našem případě bude stránka rozdělena na jednoduché menu a prostor pro zobrazení samotného obsahu. Budeme zobrazovat hlavně seznamy auditů, formuláře a výsledky graficky prezentovat. Pro výpis obsahu tak potřebujeme co největší plochu. Kdybychom zvolili layout s vertikálním menu, připravili bychom se o cenný prostor. Použití dvouúrovňového horizontálního menu nám umožní využít celé šířky zobrazovací plochy.

V poslední řadě se zaměříme na výsledný vzhled aplikace. Cílem designu není zaujmout a přilákat nové zákazníky, ale vytvořit přehledný systém s intuitivním ovládáním. Proto není potřeba vymýšlet speciální pestrobarevné kombinace, ale lze si vystačit s jednoduchým schématem o jedné či dvou barvách. Aplikace je laděna do zelených a modrých tónů s důrazem na kontrast textu s pozadím. Výsledné barvy byly zvoleny tak, aby byly pro vnímání lidského oka co nejpříjemnější. Neunavují oči a nepůsobí agresivně.

## Kapitola 5

# Implementace

### 5.1 Model životního cyklu aplikace

Na začátku celého vývoje aplikace bylo provedeno několik sezení, kde zadavatel popsal problematiku provádění auditů. Obě strany pak diskutovali získané informace a vytvořili celkovou představu o výsledné aplikaci. Mimo funkcí, které vyplývají ze specifikace stávajícího procesu provádění auditů, přidal zadavatel návrhy na další rozšíření.

Na základě těchto schůzek byla sepsána takzvaná **zadávací dokumentace**, která vzniká vždy při vývoji rozsáhlejšího softwaru. Obsahuje informace o *zadavateli aplikace, vývojáři aplikace, o požadovaných funkcích, předběžném časovém plánu a způsobu testování*. Uzavření tohoto dokumentu by mělo eliminovat situaci, kdy zadavatel během vývoje aplikace neustále rozšiřuje své požadavky na aplikaci a tím degraduje celou analýzu a předpokládanou časovou náročnost. Jakmile je tento dokument podepsán, může být zahájen vývoj aplikace.

Celý systém provádění auditů není úplně triviální. Každá část – uživatelé, audity, akce a reakce – obsahuje velké množství informací a logiky. I přes veškerou snahu nebylo možné specifikovat všechny vlastnosti do nejmenšího detailu. Proto byl použit iterativní model životní cyklu popsáný v [4]. Základem tohoto modelu je rozdělení procesu vývoje softwaru do jednotlivých iterací. Každou iteraci můžeme chápat jako posloupnost několika akcí:

1. specifikace požadavků
2. návrh
3. implementace
4. testování.

Po každé iteraci je otestována pouze určitá část aplikace. Upřesní se požadavky, aplikují se změny a pokračuje se v implementaci dalších částí. Tento postup se neustále opakuje.

Prvním úkolem bylo vytvoření statické aplikace – rozdělení aplikace na jednotlivé logické celky a zobrazení pseudoreálných dat v reálné aplikaci. Podle modelu z kapitoly 4.2 byla vytvořena databáze, obsahující tabulky s odpovídajícími atributy, cizími a unikátními klíči. Do databáze byla vložena pseudoreálná data a pomocí php propojena s uživatelským rozhraním.

Po ověření správné implementace statického obsahu byla jednotlivým částem postupně přidělována funkcionality.

Prezentace dat a celkový design aplikace byl řešen v poslední fázi implementace. Testování aplikace je popsáno v kapitole 6.



## 5.2 Přihlášení do aplikace – integrace do současného systému

### Autentizace

Počítače a servery ve firmě TRA běží na operačních systémech od Microsoftu. Ke správě síťových prvků je využívána služba **Active Directory**. Jedná se o adresářovou službu, která obsahuje například informace o počítačích, tiskárnách a uživatelských účtech. Ověřování a autorizace uživatelů je prováděna pomocí LDAP. Každý zaměstnanec, který má přístup do interní sítě TRA, vlastní pouze jeden login a heslo. Tyto údaje využívá pro přihlášení do každé aplikace. Před přístupem do aplikace je potřeba uživatele autentizovat. Vše nám pomůžou zajistit funkce jazyka php.

Postup při autentizaci je následovný:

1. Uživatel zadá do formuláře login a heslo, které užívá pro přihlášení do PC
2. Pomocí funkce `ldap_connect` se pokusíme navázat spojení s LDAP serverem
3. Je-li LDAP server dostupný, využijeme funkce `ldap_bind` a zkusíme se přihlásit pomocí zadaných přihlašovací údajů
4. V případě, že je v adresáři nalezen uživatel s danými parametry, je přihlášen do aplikace
5. Při neúspěšném přihlášení jsou funkce aplikace omezené – viz. 4.1 oprávnění guest

### Ustanovení spojení – session ID

Pokud se úspěšně přihlásíme do webové aplikace a provedeme akci, která nás přesměruje na jinou stránku aplikace, ztrácíme informaci o tom, že byl uživatel přihlášen. HTTP adresa tuto informaci neukládá. Tento problém vyřešíme pomocí php.

Ve chvíli, kdy se začne načítat libovolná stránka aplikace, je zavolána php funkce `session_start()`. Tím jsme, zjednodušeně řečeno, ustanovili spojení s aplikací. Toto spojení je identifikováno unikátním identifikačním číslem – **session ID**. Až do uzavření prohlížeče bude tato funkce generovat stejné ID. Toho využijeme tak, že při přihlášení do aplikace je uživateli přiděleno vygenerované session ID. Po každém načtení nové stránky se aplikace pokusí najít vygenerované ID v tabulce uživatelů. Pokud takový záznam najdeme, můžeme jednoznačně určit přihlášeného uživatele.

### Autorizace

Autorizace uživatele probíhá v samotné aplikaci. Seznam všech uživatelských účtů zaměstnanců firmy je uložen v globální tabulce *global users*.

Seznam uživatelů s oprávněním pro aplikaci audity je uložen v interní tabulce aplikace – konkrétně tabulka *uživatelé*.

Postup při autorizaci:

1. Po úspěšné autentizaci se pokusí aplikace vyhledat uživatele v interní tabulce *uživatelé*
2. Pokud není uživatel v aplikaci evidován, je automaticky do aplikace přidán. Informace o uživateli jsou vyhledány a zkopírovány z *global users*. V tuto chvíli má práva jako běžný uživatel. O přidělení práv musí uživatel požádat administrátora.

3. Uživatel je úspěšně přihlášen a je mu přiděleno session ID.

### Odhlášení

Doba přihlášení není nijak omezena časovým intervalem. Uživatel zůstává přihlášen do té doby, dokud se sám neodhlásí nebo dokud nezavře webový prohlížeč.

## 5.3 Správa uživatelů

Už víme, že uživatelské účty jsou spravovány v Active Directory(5.2). Každý kdo má uživatelský účet se může přihlásit do aplikace. Po přihlášení potřebujeme vědět, do kterých částí aplikace může uživatel přistupovat. K tomu slouží tabulka uživatelů, kde jsou tyto informace uchovány. Každý záznam o uživateli je vytvářen a aktualizován částečně automaticky.

Při první akci, kterou uživatel vykoná, je vytvořen záznam o tomto uživateli. První akce nemusí být nutně přihlášení do aplikace. Může se například jednat o situaci, kdy mistr nastaví nového řešitele auditu. Na výběr má z globální tabulky uživatelů. Vybere uživatele a o zbytek se postará aplikace. V případě, že uživatel neexistuje, vytvoří nový záznam a nastaví práva řešitele, jinak pouze záznam aktualizuje. Tímto způsobem se automaticky aktualizují další role uživatelů. Výjimku tvoří hlavní auditor a administrátor, které je nutné nastavit v administraci ručně. Deaktivace jednotlivých rolí je čistě v administrátorské režii.

## 5.4 Aktualizace vztahu buňka-mistr

Kontrolovat buňku po buňce a její přiřazení k mistrovi by bylo velice nepraktické. Ideálním případem by bylo propojit aplikaci s globální tabulkou, která tyto informace obsahuje. Bohužel, žádná taková v databázi není. Respektive její aktuálnost nikdo neudrží. Jediným zdrojem ověřených informací je soubor ve formátu *xls*. Soubor obsahuje daleko více informací než potřebujeme. Nás bude zajímat sloupec označený “Kód stroje” pro název buňky a sloupec “Nový mistr” pro jméno mistra. Pořadí a počet sloupců se může libovolně měnit. Názvy sloupců pro označení buňky a mistra musí zůstat stejné. Před nahráním do aplikace uložíme soubor ve formátu *csv*(*oddělený středníkem*).

### Formát CSV – *Comma-separated values*.

Tento formát usnadňuje práci s tabulkovými daty. Každá hodnota sloupce je oddělena buď čárkou, středníkem, nebo tabulátorem. Oddělovací znak závisí na způsobu uložení. [7]

My budeme využívat středník, abychom nemuseli řešit případy, zda je čárka součástí hodnoty, nebo oddělovací symbol.

K odeslání souboru do aplikace využijeme HTML formulář a metodu POST. Nahráný soubor pak jednoduchým cyklem rozdělíme na jednotlivé řádky. Každý řádek má stejný počet sloupců a je ukončen znakem konce řádku. Nejprve analyzujeme první řádek. Pozice stringu “Kód stroje” a “Nový mistr” nám určuje indexy pole, na kterých jsou uloženy názvy buněk a jména mistrů. Ostatní sloupce nás zajímat nebudou. Postupným procházením řádků aktualizujeme informace v databázi. Ošetřujeme vstupy, kdy buňka nebo uživatel v databázi neexistují a případy, kdy chybí jedna ze dvou informací. O výsledku provedení skriptu je uživatel informován výpisem na obrazovku (včetně chybových hlášení).

## 5.5 Aktualizace otázek a aktualizace typu auditu

Každá otázka a každý typ auditu může mít několik verzí. Platná smí být pouze vždy jenom jedna. Otázky nemusí být ve stejné verzi, jako je verze typu. Jedinou podmínkou je, že musí být platné.

### Aktualizace otázky

Hlavní auditor má možnost upravit konkrétní otázku. Může změnit kategorii a text otázky. Změna může mít tyto varianty:

- Promítne v celém systému – otázka se změní také v již provedených auditech.
- Promítne se pouze do budoucna – provedené audity zůstanou nezměněny. V takovém případě je potřeba vytvořit novou verzi otázky a až poté otázku upravit.

Vytvořením nové verze otázky se automaticky zneplatní verze předchozí – tato verze již nebude dále nabízena. Vytvořením nové verze otázky se také automaticky vytváří nová verze všech typů, ve kterých se daná otázka nachází. Předešlé verze typů se také zneplatní.

### Aktualizace auditu

Hlavní auditor má možnost upravit konkrétní typ auditu. Může změnit pořadí otázek, odebrat otázku, přidat otázku, změnit název. Stejně jako u aktualizace otázky může být změna celoplošná a nebo se projeví pouze v auditech, které budou provedeny. Verze všech otázek se automaticky aktualizuje na novější verzi a starší se zneplatní.

#### 5.5.1 Otázka vyskytující se ve více formulářích

Pokud by se otázka vyskytovala pouze v jednom auditu, bylo by vše bezproblémové. To ovšem není náš případ. Jedna otázka může totiž patřit do více typů. Musíme dávat tedy velký pozor, abychom vždy aktualizovali správně všechny otázky a všechny typy. I když se v praxi využívají zatím pouze dva typy, počítá se do budoucna s vytvořením dalších typů. Tento poznatek musíme zvážit při vytváření algoritmu pro správnou aktualizaci. Nejlépe si vzniklou situaci popíšeme na následujícím příkladu.

*Máme 4 typy auditu: Ax, Bx, Cx, Dx a k nim příslušné otázky. V závorce je uvedena verze typu, respektive otázky. Obrázek 5.1.*

TYP	OTÁZKY		
Ax (ver. 1)	K(1)	L(1)	M(1)
Bx (ver. 2)	O(2)	P(1)	W(1)
Cx (ver. 1)	S(1)	T(3)	V(1)
Dx (ver. 1)	K(1)	U(1)	W(1)

Obrázek 5.1: Stav jednotlivých typů auditu před aktualizací Ax.

1. Aktualizujeme typ Ax(1).

2. Vytvoří se nová verze Ax(2) a stará verze Ax(1) se zneplatní. To samé se provede s otázkami tohoto typu. K(1) je povýšena na novější verzi K(2) a stará verze K(1) je zneplatněna. Stejným způsobem jsou aktualizovány zbývající otázky.

Pokud by byl na tomto místě algoritmus ukončen, zůstala by v systému nekonzistentní data. Typ Dx(1) by obsahoval otázku K(1), která je již neplatná. Žádný aktivní typ nesmí obsahovat neplatnou otázku. Správně se musí aktualizovat celý typ Dx(1) na Dx(2) a tím dojde k aktualizaci všech jeho otázek. Nastává další konflikt s otázkou W(1). Ta už byla v předchozím kroku aktualizována na W(2) a W(1) byla zneplatněna. Musíme tedy aktualizovat i typ Bx(2) na novější verzi Bx(3).

Tímto způsobem je nutné postupně projít a aktualizovat všechny propojené typy. Výsledný stav po aktualizaci je na obrázku 5.2

TYP	OTÁZKY		
Ax (ver. 2)	K(2)	L(2)	M(2)
Bx (ver. 3)	O(3)	P(2)	W(2)
Cx (ver. 1)	S(1)	T(3)	V(1)
Dx (ver. 2)	K(2)	U(2)	W(2)

Obrázek 5.2: Stav jednotlivých typů auditů po aktualizaci Ax.

### Algoritmus

1. najdi všechny aktivní otázky vyskytující se ve více auditech (*dále “multiotázka”*), ke každé z nich přiřaď množinu auditů, ve kterých se nachází
2. existuje-li alespoň jedna otázka v typu, který právě aktualizujeme, znamená to, že aktualizace ovlivňuje i další typy
3. postupně najdeme všechna propojení a vytvoříme seznam typů k aktualizaci
4. ze seznamu typů zjistíme seznam všech otázek
5. povýšíme verze nalezených typů a otázek a staré verze deaktivujeme

```

if(multiotázka in aktualizovaný_typ){
    přidej typy které ovlivňuje do typů k aktualizaci;

    while (průchod=1 <= počet_multiotázek){

        foreach (multiotázky){ // projdi každou multiotázku
            if(typy které ovlivňuje in typy k aktualizaci)
                aktualizuj seznam typů k aktualizaci;
            unset porovnaná_otázka;
        }
    }
} // konec if
ze seznamu typů vyber všechny otázky;

```

```
vytvoř.nové_verze_typů_a_otázek;  
deaktivuj.předchozí_verze;
```

## 5.6 Notifikace prostřednictvím emailu

Typy notifikací:

- po provedeném auditu je mistrovi zaslán výsledek kontroly
- na konci týdne je každému uživateli, který neplní své úkoly, zasláno upozornění týkající se konkrétních auditů; u každé položky je odkaz do aplikace přímo na vykonání konkrétní akce
- upozornění na naplánovaný audit; zasláním emailu je automaticky naplánována událost do Outlook Express kalendáře

Zasílání emailů umožňuje funkce `php`.

`mail(to,subject,message,headers,parameters)`

- *to* – emailová adresa příjemce v platném formátu
- *subject* – předmět emailu
- *message* – obsah zprávy
- *headers* – v hlavičce jsou uvedeny informace o odesílateli, o typu kódování; definuje typ zprávy atp. [6]

K odesílání emailů využijeme MIME <sup>1</sup> standard, který obohacuje klasický emailový formát o nové možnosti [5].

- podpora textu psaného v různých znakových sadách
- podpora zasílání příloh – obrázky, videa, audio, kalendářní událost
- zpráva může být různorodá – jednotlivé typy můžeme mezi sebou kombinovat

Typ zprávy je určen v hlavičce příkazem:

**Content-Type: type/subtype**

Konkrétní příklady:

```
Content-Type: text/html;           // text ve formátu HTML  
Content-Type: text/calendar;       // text ve formátu calendar  
Content-Type: image/jpeg;          // obraz ve formátu jpeg  
Content-Type: multipart/alternative; // vícedílná zpráva obsahující více typů
```

Podrobnější specifikace je v [3].

---

<sup>1</sup>Multipurpose Internet Mail Extensions – víceúčelová rozšíření internetové pošty.

## 5.7 Automatické plánování událostí do kalendáře Microsoft Outlook Express

Pro komunikaci, organizaci času, schůzek, událostí atp. je využíván Microsoft Outlook Express (dále “OE”). Zaměstnanci si do OE zapisují velké množství informací. Jednotlivé události plánují do svého OE kalendáře. Díky tomu je v systému vidět, kdy je daný uživatel dostupný. My doplníme kalendář OE o další informaci. Využijeme standard *iCalendar* a zakomponujeme plán auditů do individuálního kalendáře každého uživatele.

### 5.7.1 iCalendar

iCalendar je formát, sloužící pro výměnu a správu dat kalendáře – událost, schůzka, poznámka, úkol, alarm apod. Byl vytvořen ze starší verze vCalendar.

Dnes je iCalendar využíván v mnoha počítačových, webových aplikacích a systémech a patří mezi nejrozšířenější a nejpoužívanější standardy pro práci s daty kalendáře. Podporu najedeme například u Outlook Calendar, Google Calendar, Windows Calendar a dalších.

Data formátu iCalendar jsou uloženy v souboru jako prostý text. Základní stavební jednotku tvoří komponenta **iCalendar object**. Zjednodušeně se dá říci, že se jedná o jednu položku v kalendáři. Jsou v ní uvedeny všechny potřebné informace a vlastnosti. Sekvence jednoho nebo více těchto objektu vytváří celek – kalendář, který se označuje jako *Calendar*ing Core Object(dále CCO).[2]

#### STRUKTURA CCO

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//hacksw/handcal//NONSGML v1.0//EN
....
iCalendar object
iCalendar object
...
END:VCALENDAR
```

Příkazy *BEGIN:VCALENDAR* a *END:VCALENDAR* deklarují tělo CCO. Vše, co se nachází uvnitř, je tělo souboru – *icalbody*. V těle jsou definovány atributy kalendáře a jednotlivé iCalendar objekty.

Povinnými atributy CCO je použitá verze kalendáře, jednoznačné id a alespoň jeden iCalendar object.

#### iCalendar object

Výčet jednotlivých iCalendar objektů[2]

- Událost (VEVENT)
- Úkol (VTODO)
- Deník (VJOURNAL)
- Volný a obsazený čas (VFREEBUSY)

- Alarm (VALARM)
- Časová zóna (VTIMEZONE)

Všechny tyto komponenty mají stejnou strukturu a syntaxi. Tělo je deklarováno analogicky jako u CCO příkazy *BEGIN* a *END*. Každá jedna vlastnost komponenty musí být uvedena uvnitř této oblasti a na samostatném řádku.

V naší aplikaci budeme využívat VTIMEZONE a VEVENT.

#### VTIMEZONE – časová zóna

Nastavení časové zóny je kritické pro správné naplánování události do kalendáře. Čas a datum odesíláme v `unix timestamp`<sup>2</sup> formátu. Pokud aplikace Microsoft Outlook Express neobdrží informace o časovém razítku, nemusí nastavit správné datum a čas. Lépe řečeno nastaví jinou časovou zónu, než kterou potřebujeme. Protože plánujeme událost na celý týden od pondělí 00:00 do neděle 23:59, musí být konverze přesná, aby rozsah události nezasahoval do dvou týdnů.

#### VEVENT – událost

Pro vytvoření události v kalendáři využijeme komponentu VEVENT. Povinnými atributy jsou DTSTART a DTEND určující začátek a konec události. Dalším povinným atributem je jednoznačný identifikátor. Na základě identifikátoru můžeme znovu přistupovat k dané události a editovat nebo dokonce smazat tuto událost. Jako organizátora události nemusíme uvádět reálnou osobu, ale stačí uvést pouze textové označení jako “Aplikace AUDIT” apod. Email organizátora zvolíme příhodně tak, aby bylo patrné, že jde o automatickou informační zprávu, na kterou není potřeba odpovídat – například *neodpovidat@trcz.thk.com*. Do předmětu události vyplníme typ auditu a buňku. Na událost můžeme upozornit určitý časový interval před jejím začátkem. K těmto účelům slouží komponenta VALARM. Posledním důležitým parametrem je status události. Od toho se odvíjí i stav uživatele v době konání dané události. Je-li status události “zanepřázdňen”, je v danou dobu “zanepřázdňen” i uživatel. Protože naplánovaný audit zabírá v kalendáři celý týden, nastavíme status události na “Volno”. Pokud bychom nenastavili tento parametr, byl by uživatel evidován po celý týden jako “zanepřázdňený”.

Příklad CCO:

```
BEGIN:VCALENDAR
PRODID:\APL-AUDITY-THK-DACICE-EN
VERSION:2.0
METHOD:REQUEST

BEGIN:VTIMEZONE
TZID:Europe-Prague
BEGIN:STANDARD
DTSTART:20091001T020000
RRULE:FREQ=YEARLY;INTERVAL=1;BYDAY=4SU;BYMONTH=10
TZOFFSETFROM:+0200
```

---

<sup>2</sup>Unix timestamp – počet sekund uplynulých od (UTC) 00:00:00 1. ledna 1970

```

TZOFFSETTO:+0100
TZNAME:EST
END:STANDARD
END:VTIMEZONE

BEGIN:VEVENT
ORGANIZER:Aplikace
ATTENDEE:auditor@thk.com
UID:20170106T000000valenr@thk.com
DTSTAMP:20170106T000000
DTSTAR:20170306T000000
DTEND:20170312T235900
X-MICROSOFT-CDO-BUSYSTATUS: FREE
SUMMARY: Plan auditu
LOCATION: LPA1 na~A21

END:VEVENT
END:VCALENDAR

```

Způsob odesílání emailů je popsán v kapitole 5.6. Před odesláním nastavíme hlavičku emailu:

Content-Type: text/calendar;name="meeting.ics";method=REQUEST

Jako obsah zprávy odešleme definovaný CCO. Výsledkem je naplánovaná událost v kalendáři OE (viz.5.3)

	28 úterý	1 středa	2 čtvrtek
	LPA1 AUDIT na bunce C33; BUNKA C33; Aplikace Audit		
	LPA1 AUDIT na bunce H31; BUNKA H31; Aplikace Audit		
7 <sup>00</sup>			
8 <sup>00</sup>			
9 <sup>00</sup>	Aplikace LPA, 5S a HSE auditů DaciceCZ-Ucebna-Mezipatro Iva Hopianova		Porada před školením ; DaciceCZ-Ucebni
10 <sup>00</sup>			Aplikace LPA, 5S a HSE auditů; DaciceCZ

Obrázek 5.3: Naplánované události v kalendáři Outlook Express



## 5.8 Grafická prezentace výsledků

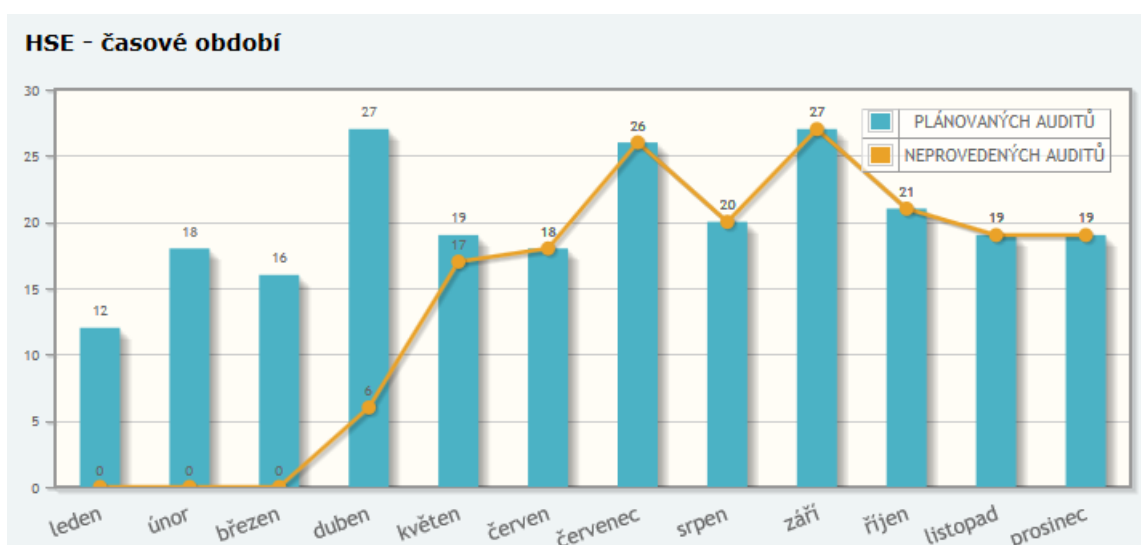
Přehled vyhodnocovaných oblastí je uveden v kapitole 2.3.4. Prezentace výsledků je nejen v textové podobě, ale i v podobě grafické. Pomocí SQL dotazů získáme z databáze potřebné informace, které následně vypíšeme. Stejná data znázorníme graficky pomocí sloupcových grafů a přímek na ose x a y. Budeme tak vytvářet dvourozměrné grafy.

Základem grafu je jedna takzvaná složka, kterou si můžeme představit jako hodnotu zkoumané oblasti a parametr, ke kterému se hodnota váže.

*Př.: nevyhovující audity za určitou dobu; nevyhovující audit v závislosti na mistrovi.*

Hodnota zkoumané oblasti se váže na osu y. Parametr je spojen s osou x. Každá složka je vykreslena specifickým způsobem – sloupec, bod, oblast, čtverec. Soubor několika složek vytváří graf.

Výsledné zobrazení může obsahovat kombinaci několika grafů. Musí být ale závislé na stejném parametru.



Obrázek 5.4: Graf znázorňující počet naplánovaných a neprovedených auditů za každý měsíc

Pro práci s grafy budeme používat plugin jqplot (3.4.2). Plugin není potřeba složitě instalovat. Stačí ho pouze stáhnout z oficiálních webových stránek, umístit do adresářové struktury aplikace a cestu ke složce definovat v HTML souboru.

Vykreslení grafu obstará funkce `jqplot`, která vyžaduje celkem tři parametry.

`\$.jqplot(target,data,options)`

- **target** – určuje prostor, do kterého je graf vykreslen. V HTML definujeme blokový prvek s jednoznačným identifikátorem, který funkci předáme.
- **data** – hodnoty osy y
- **options** – objekt, definující vzhled a funkcionalitu grafu; uvnitř nastavujeme hodnoty osy x

Options objekt se skládá z dalších dílčích objektů. Základními komponenty jsou *axis*, *series*, *grid*, *legend*. Uvnitř je soubor vlastností a funkcí popisující chování a vzhled daného

objektu. Definovat můžeme barvu, rozměry, umístění objektu, průhlednost, typ vykreslení, styl animace, rychlost animace, chování při interakci s okolím (např.: kliknutí myši) a další. To je jen malý výčet všech možností. Kompletní seznam vlastností a funkcí je dostupný v dokumentaci na oficiálních stránkách.<sup>3</sup>

Ukázka funkce jqplot:

(pozn.: kód slouží pro ilustraci, výčet všech vlastností není úplný)

```
$.jqplot("id_graf1", [[3,4,5],[1,2,3]], {
    animate: true,
    series:[
        { // výčet vlastností první datové serie
            renderer: $.jqplot.BarRenderer,
            rendererOptions: {
                ...
            }
        },
        { // výčet vlastností druhé datové série
            ...
        }
    ],
    axes: {
        xaxis: { // definice v~x-ové osy
            ticks: ['leden','únor','březen']
            ...
        },
        yaxis: { // definice y-ové osy
            ...
        }
    }
});
```

Popis kódu:

Konečný výsledek bude vykreslen do oblasti dokumentu, která odpovídá oblasti *id\_graf1* v HTML souboru. Jedná se kombinaci dvou grafů složených do jednoho. Pole hodnot *[3,4,5]* určuje y-ové hodnoty prvního grafu, pole *[1,2,3]* hodnoty grafu druhého.

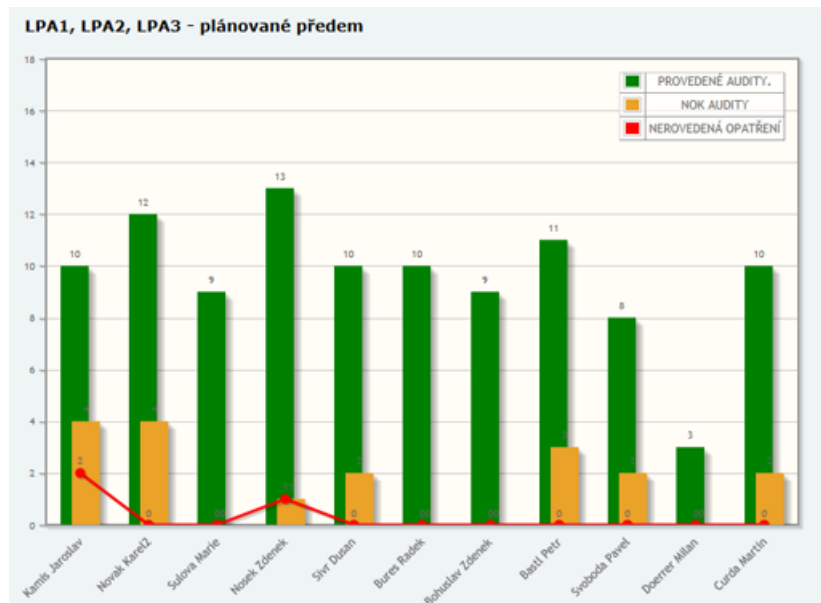
Oba grafy musí být závislé na stejném parametru – hodnoty osy x, které jsou definovány v objektu *axes*, konkrétně *xaxis*. Počet x-ových parametrů musí být stejný jako počet y-ových hodnot.

V objektu *series* jsou nastaveny vlastnosti jednotlivých grafů. Vlastnost *renderer* nastavuje způsob vykreslení. Vykreslení prvního grafu je sloupcové. Pokud není tato vlastnost uvedena, je použita výchozí hodnota – jednotlivé složky grafu jsou reprezentovány jako bod a jsou navzájem spojeny přímkami.

Podobný výsledek je ilustrován na obrázku 5.4.

---

<sup>3</sup>[www.jqplot.com/docs](http://www.jqplot.com/docs)



Obrázek 5.5: Graf znázorňující provedené audity, NOK audity, nevyřešené audity v závislosti na mistrovi.

## 5.9 Export dat do CSV

Všechna data jsou uložena v databázi a jsou dostupná přes rozhraní naší aplikace. Pokud s nimi chceme dále organizovaně pracovat mimo prostředí aplikace, pouhé kopírování textu z webového prohlížeče není příliš efektivní. Pro následnou práci s daty budeme využívat formát CSV, který je popsán v kapitole 5.4. Zadavatel předem specifikoval data, která mají být exportována pro další zpracování. Na základě těchto požadavků byl vytvořen na tyto data takzvaný pohled.

**Pohled** je virtuální tabulka, která je výsledkem SQL dotazu. Umožňuje propojit více tabulek do jedné virtuální a s tou pak pracovat jako s tabulkou klasickou.

Pro účely exportu byla vytvořena virtuální tabulka **EXPORT**. Každý řádek odpovídá jednomu auditu. Exportování se provádí ze stránky s výpisem všech auditů. Filtrem vybereme, které audity chceme vyexportovat. Chceme-li kompletní seznam, necháme filtrování vypnuté. Kliknutím na tlačítko *exportovat* zavoláme skript, po jehož úspěšném dokončení nám nabídne webový prohlížeč exportovaný soubor ve formátu CSV ke stažení.

K přenosu HTML souboru se používá HTTP protokol, který pracuje na aplikační vrstvě. Hlavičku nastavíme pomocí php funkce `header()` následovně:

```
header("Content-type: text/csv");
header("Content-Disposition: attachment; filename=file.csv");
```

Těmito příkazy jsme nastavili typ obsahu na "text/csv", výpis obsahu do souboru a název souboru "file.csv". Provedeme SQL dotaz nad tabulkou **EXPORT**. Pomocí cyklu procházíme data a vypisujeme je podle pravidel CSV. Každý sloupec oddělíme středníkem a každý řádek ukončíme znakem konce řádku.

Po úspěšném dokončení skriptu nám webový prohlížeč automaticky nabídne exportovaný soubor ke stažení.

## 5.10 Ošetření výjimečných případů

Podle diagramu případu užití je aplikace rozdělena na jednotlivé funkční celky. Přístup do těchto částí závisí na právech uživatele, která zjistíme funkcí `getUserRights` ihned při spuštění aplikace. Parametrem funkce je session id (*kapitola 5.2*). Díky této informaci identifikuje přihlášeného uživatele a do globální proměnné `USER_RIGHTS[ ]` uloží jeho práva. Tato proměnná je pak testována před výpisem každé položky menu a výpisem příslušného obsahu. Uživatel se tedy běžným procházením aplikace nemá k neoprávněnému obsahu jak dostat, protože zjednodušeně řečeno neví, že existuje.

Musíme ošetřit i takové situace, kdy se uživatel snaží vstoupit k neoprávněným datům přes přímí link. Protože se jedná o interní aplikaci tak neuvažujeme, že by se o to pokoušeli zaměstnanci vědomě. Protože na jednom počítači může s aplikací pracovat více lidí s různým oprávněním a webový prohlížeč si ukládá historii navštívených stránek, může čas od času k této situaci dojít. Stejně tak musíme řešit situaci, kdy se uživatel pokusí provést některou nevalidní akci. Typ akce, která má být provedena je posílán v URL adrese.

```
page=fillaudit&id_audit=636
```

Tato informace uvedena v URL adrese odkazuje aplikaci na provedení auditu číslo 636. Pokud zaměníme číslo auditu, aplikace musí rozpoznat, zda má uživatel právo příslušný audit vyplnit. Stejně tak musí rozpoznat, zda se pokouší provést validní akci. V žádném případě nemůže uživatel vyplnit ten samý audit dvakrát. Těchto výjimek je v aplikaci celá řada. Nelze potvrdit již potvrzený audit, řešit vyřešený, řešit neprovedený atp.

## Kapitola 6

# Testování a uvedení do provozu

Testování aplikace se modelu životního cyklu, který je popsán v kapitole 5.1. Testování probíhalo ve dvou fázích. Základní testování prováděli hlavní auditoři iterativně v průběhu implementace. Jakmile byla aplikace dokončena, proběhlo zaškolení uživatelů a mohla tak být spuštěna druhá fáze testování, která proběhla ve formě zkušebního provozu. Do této fáze byli zapojeni všichni zaměstnanci, kterých se provádění auditů týká.

### 6.1 Testování aplikace v průběhu vývoje

Hlavní auditoři rozumí nejvíce celé problematice a znají celý proces ze všech uživatelských pozic. Po implementaci určité části aplikace byla tato část odprezentována a okomentována tak, jak chápe celou situaci programátor. Následně proběhlo testování, analýza a potvrzení nebo vyvrácení správnosti implementované části. Tento průběh se opakoval v několika iteracích. Zpočátku bylo testování pouze teoretické ve formě diskuse. Postupem času se utvořilo statické jádro aplikace a byla implementována funkcionality a logika operací nad tímto jádrem. Další testování bylo už náročnější a obsahovalo teoretickou i praktickou část. Během testování bylo postupně odhaleno několik nedostatků ve specifikaci aplikace:

**Problém:** Typy auditů

**Domněnka:** Existují celkem 4 typy auditů – HSE, LPA1, LPA2, LPA3. Každý typ je chápán jako samostatný celek obsahující jiné otázky.

**Skutečnost:** Existují pouze dva typy – HSE a LPA. Číselné označení určuje pouze úroveň auditu. Jinak se jedná o stejný typ. Struktura formuláře je u všech úrovní stejná. Úroveň auditu navíc určuje auditora, který může audit provádět.

**Důsledek:** Tato nesrovnalost odhalila dosud neznámou informaci – existuje více úrovní auditorů. Na základě tohoto zjištění byla upravena tabulka s uživateli.

**Problém:** Povinné atributy plánovaného auditu

**Domněnka:** Přidělování auditorů je čisté v kompetenci hlavních auditorů. Každý naplánovaný audit musí mít jasně přiděleného auditora.

**Skutečnost:** Domněnka platí pouze pro LPA audit. U HSE je při plánování auditu jako auditor určen mistr, který nastavuje ručně auditora ze svého týmu. Nastavení auditora je mu umožněno dva týdny před plánovaným provedením.

**Důsledek:** Vytvoření nové funkcionality – nastavení auditora mistrem.

**Problém:** Nejasné verze typů a otázek

**Domněnka:** Při vytvoření nové verze auditu se aktualizují pouze otázky, u nichž došlo ke změně jejich znění.

**Skutečnost:** Touto situací si nebyli jisti ani samotní zadavatelé. Nakonec byla zvolena varianta, kdy vytvoření nové verze typu aktualizuje automaticky na novou verzi i všechny příslušné otázky.

**Důsledek:** Upravení algoritmu.

**Problém:** Řešitel auditu

**Domněnka:** Každý audit má jednoho řešitele.

**Skutečnost:** Audit může řešit více řešitelů.

**Důsledek:** Upravení databáze. Vytvoření nové tabulky – řešitel.

V průběhu analýzy částečné aplikace vznikaly další návrhy na nové funkce. Ty nejnepatnější jsou zmíněny v závěrečné kapitole.

## 6.2 Zaškolení uživatelů

Před zkušebním provozem bylo potřeba zaškolit všechny uživatele, kteří budou aplikaci využívat. Školení byla rozdělena na tři skupiny podle rolí uživatelů. Zvlášť byl s aplikací seznámen management, zvlášť technologové a zvlášť mistři. Každý mistr si pak řídil školení svého týmu samostatně.

## 6.3 Zkušební provoz

Zkušební provoz byl naplánován na únor 2017. Cílem bylo otestovat aplikaci v co nejširším možném měřítku. Oficiálně se stále pracovalo starým systémem, ale poprvé byla aplikace přístupna všem uživatelům. Aby si mohl každý vyzkoušet běh aplikace v neomezené míře, byly vytvořeny provizorní účty pro každou jednotlivou roli. (4.1). Harmonogram zkušebního provozu byl rozdělen na čtyři etapy. První a třetí týden si mohli pracovníci vyzkoušet práci s aplikací a hledat případné chyby. Druhý a čtvrtý týden byl vyčleněn na odstranění nalezených problémů.

Během zkušebního provozu nebyly odhaleny žádné fatální problémy. Jeden závažnější problém se objevil při plánování nového auditu. V důsledku chybné syntaxe JavaScriptu neprobíhala kontrola validních dat odesílaného formuláře. Tím mohli vznikat audity bez přiděleného auditora a dalších informací. Taková situace neodpovídala specifikaci a způsobovala v aplikaci problémy. Bylo také zjištěno špatné nastavení oprávnění LPA1 auditorů. Nemohli tak přistupovat k auditům první úrovně. Další odhalené chyby byly spíše drobnějšího charakteru. Při provedení auditu se například neukládala hlavička formuláře u LPA auditů. U filtrování dat nefungovalo filtrování podle buňky. Všechny tyto nedostatky byly odstraněny a vývoj první verze aplikace mohl být ukončen.

1. března 2017 byla oficiálně spuštěna oficiální verze aplikace Audity 1.0.

## Kapitola 7

# Závěr

Výsledkem této bakalářské práce je webová aplikace umožňující snadnou správu prováděných auditů. Podnětem pro vznik této aplikace byl neefektivní způsob, který firma pro tuto činnost využívala. Hlavním problémem předešlého systému byla značná nepřehlednost. Provádění kontrol mělo za cíl odhalit nejčastější problémy a pomoci předcházet jejich vzniku. Místo toho, aby byly problémy odstraněny, přispěl chaotický systém ke vzniku dalších neshod. Někteří zaměstnanci neměli přehled o aktuální situaci a často tak nevědomě neplnili své povinnosti. Naopak jiní se pokoušeli zbavit své odpovědnosti vymlouváním se na nepřehledný systém – “On mi to nikdo neřekl.”, “Jak jsem to měl vědět?” atp.

Ostrý provoz běží teprve dva a půl měsíce, ale i na základě tohoto krátkého časového úseku můžeme konstatovat, že náhrada starého systému touto webovou aplikací byl krok tím správným směrem.

V první řadě pomohla aplikace zpřehlednit celou situaci. Veškeré informace spojené s kontrolami jsou dostupné z jednoho zdroje. Každý uživatel je o veškerém dění včas informován. Ve svých auditech vidí přesně to, čemu má co nejdříve věnovat svoji pozornost. Nemůže se vymlouvat, že mu někdo něco neřekl, nebo že nevěděl, co má dělat. Hned na první pohled je v aplikaci vidět, kdo plní své úkoly svědomitě a kdo naopak své povinnosti nedodržuje.

Velkým přínosem je tato aplikace zejména pro hlavní auditory, kteří dohlíží na celý systém a každý měsíc vyhodnocují sledované ukazatele. Nejvíce ocení automatické generování grafů a výpis statistik. Odpadá jim tak starost vyhledávat naskenované audity na serverovém úložišti. Tyto data musely nejprve zpracovat do textové podoby a poté graficky vyhodnotit. Grafy jsou navíc obohaceny o funkci filtrování podle zadaných parametrů. Zjistit ve starém systému evidence dat například informaci jako *nejčastější problémy na buňce A2 za poslední dva měsíce* bylo prakticky nemožné. Uložená data také nebyla dost často kompletní, protože je zaměstnanci zapomínali na server nahrát. Někteří neprováděli audity včas. Někteří neřešili problémové audity. Neustálá urgence těchto zaměstnanců byla velice únavná.

K datu 1. března 2017 byl zahájen ostrý provoz. Vývoj aplikace ale pokračuje dál. Již v tuto chvíli je vytvořen přibližný časový plán pro druhou fázi vývoje. Do konce měsíce června jsou zaznamenávány požadavky a připomínky od všech uživatelů. Následně se zpracují a proběhne jejich analýza, implementace, testování a nakonec nasazení druhé verze. Již během vývoje postupně vznikaly náměty na další rozšíření. Většinou se jedná o designové drobnosti, které zpřijemňují práci s aplikací. Mezi ty důležitější patří například vytvoření vícejazyčné prostředí. Podobný systém je využíván i v dalších závodech THK, například v Německu, a aplikace by tak našla své uplatnění i tam. Nová funkcionalita potvrzená pro

další verzi se týká plánování auditů. Konkrétně vytváření plánu pro celý rok. Ten obsahuje přes 300 auditů, které podléhají určitým pravidlům. Vypracovat takový plán je časově náročné, proto bude tato činnost prováděna v další verzi již automaticky.

Při vývoji této aplikace jsem si ověřil, že precizní a dobrá analýza je základem úspěchu. Nekvalitní analýza nevzniká špatně odpovězenými otázkami, ale špatně formulovanými otázkami. I přes veškerou snahu popsat vše do nejmenších detailů se objevila řada nejasností, které narušili plynulý vývoj aplikace. Výhodou bylo, že implementace vznikala ve stejném prostředí jako se nacházel i zadavatel. Upřesnění drobného detailu pak mohlo být konzultováno téměř okamžitě bez nutnosti plánovat speciální externí schůzky.



# Literatura

- [1] *History of PHP* . [Online; navštíveno 26.04.2017].  
URL <http://php.net/manual/en/history.php.php>
- [2] B. Desruisseaux, E.: *Internet Calendaring and Scheduling Core Object Specification (iCalendar)* . Září 2009, [Online; navštíveno 09.05.2017].  
URL <https://tools.ietf.org/html/rfc5545>
- [3] Freed, N.; Borenstein, N.: *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* . Listopad 1996, [Online; navštíveno 07.05.2017].  
URL <https://tools.ietf.org/html/rfc2045>
- [4] Křena, B.: *Úvod do softwarového inženýrství*. FIT VUT v Brně, Prosinec 2010, [Online; navštíveno 29.04.2017].  
URL [https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IUS-IT/texts/IUS\\_opora.pdf](https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IUS-IT/texts/IUS_opora.pdf)
- [5] Moore, K.: *MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text* . Listopad 1996, [Online; navštíveno 01.05.2017].  
URL <https://tools.ietf.org/html/rfc2047>
- [6] P. Resnick, E.: *Internet Message Format* . Duben 2001, [Online; navštíveno 03.05.2017].  
URL <https://tools.ietf.org/html/rfc2822>
- [7] Shafranovich, Y.: *Common Format and MIME Type for Comma-Separated Values (CSV) Files* . Říjen 2005, [Online; navštíveno 02.05.2017].  
URL <https://tools.ietf.org/html/rfc4180>
- [8] Tutorials, W. O. W.: *CSS Introduction*. 2017, [Online; navštíveno 27.04.2017].  
URL [https://www.w3schools.com/css/css\\_intro.asp](https://www.w3schools.com/css/css_intro.asp)
- [9] Tutorials, W. O. W.: *HTML Introduction*. 2017, [Online; navštíveno 26.04.2017].  
URL [https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)
- [10] Zendulka, J.: *Databázové systémy* . FIT VUT v Brně, Červenec 2006, [Online; navštíveno 02.05.2017].  
URL [https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IDS-IT/texts/IDS\\_predn.pdf](https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IDS-IT/texts/IDS_predn.pdf)

# Přílohy

# Příloha A

## Obsah CD

- `/actions` – zdrojové soubory pro exportování a notifikaci emailem
- `/forms` – původní formuláře ve formátu xls
- `/lib`
  - `/lib/doc` – technická zpráva ve formátu PDF
  - `/lib/css` – zdrojové CSS soubory
  - `/lib/charts` – zdrojové soubory grafů
  - `/lib/img` – obrázky
  - `/lib/inc` – konstanty a globální funkce
  - `/lib/jqplot` – zdrojové soubory pluginu jqplot
  - `/lib/js` – zdrojové soubory javascriptu
- `/sql_scripts` – obsahem složky je skript pro vytvoření databáze
- `/` – zdrojové soubory aplikace

### Konfigurační soubory

1. `/sql_scripts/createDB.sql` – skript pro vytvoření MS-SQL databáze; obsahuje reálná data a výsledky k datu 10.5.2017
2. `/lib/inc/const.php` – konfigurační soubor pro propojení aplikace s databází

## Příloha B

# Rozdělení uživatelského menu

- Přihlášení
- Administrace
  - Uživatelé – *správa uživatelského oprávnění*
  - Buňky – *přidání nové buňky; přiřazení mistra k buňce*
  - Kategorie – *editace kategorií otázek*
  - Otázky – *editace otázek*
  - Typ závady – *editace typu závady*
  - Závada-otázka – *přiřazení závady k otázce*
  - Typy auditu – *editace a aktualizace typů auditu*
  - Editace auditů – *editace provedených auditů*
- Plánování
  - Naplánovat audit – *plánování auditů*
  - Upravit plán auditů – *změna naplánovaných auditů*
- Audity
  - Aktuální audity – *přehled operací, které má uživatel provést*
  - Moje audity – *přehled všech auditů uživatele*
  - Nastavit řešitele – *přiřazení řešitele k problémovému auditu*
  - Zobrazit všechny audity – *přehled všech auditů v databázi*
- Statistiky a grafy
  - Provedené / plánované – *textové a grafické reporty*
  - Provedené / NOK / neopravené – *textové a grafické reporty*
  - Nejproblémovější otázky – *textové a grafické reporty*

# Příloha C

## Ukázka aplikace

AUDITYSTATISTIKY a GRAFYDusan Sivrv(odhlásit)

Aktuální audityMoje audityNastavit řešiteleZobrazit všechny audity

### Aktuální audity

Audity HSE a nastavení auditorů.

#	TYP	BUŇKA	MISTR	AUDITOR	VÝSLEDEK	STAV %	ŘEŠITEL	NAPLÁNOVÁN	VYBRAT AUDITORA	NASTAVIT
1	HSE	H38	SIVRD	NEUVIZ		NAPLÁNOVANÝ		22.05.2017	Neuvirt Zdenek	Nastavit auditora

### Audity k provedení

#	TYP	BUŇKA	MISTR	AUDITOR	VÝSLEDEK	STAV %	ŘEŠITEL	NAPLÁNOVÁN	PROVEDEN	VYŘEŠEN	UZAVŘEN	DETAIL
1	LPA1	A12	KAMISJ	SIVRD		NAPLÁNOVANÝ		18.05.2017	PROVÉST		NE	DETAIL
2	LPA2	A13	KAMISJ	SIVRD		NAPLÁNOVANÝ		18.05.2017	PROVÉST		NE	DETAIL
3	LPA2	A17	KAMISJ	SIVRD		NAPLÁNOVANÝ		19.05.2017	PROVÉST		NE	DETAIL


### Audity k vyřešení

#	TYP	BUŇKA	MISTR	AUDITOR	VÝSLEDEK	STAV %	ŘEŠITEL	NAPLÁNOVÁN	PROVEDEN	VYŘEŠEN	UZAVŘEN	DETAIL
1	LPA1	H34	SIVRD	HAVLIMI	86%	ŘEŠÍ SE(86%)	KOLMAN BOHUSLAV	11.05.2017	11.05.2017	VYŘEŠIT	NE	DETAIL

### Audity k potvrzení

Aktuálně nemáte žádné audity k potvrzení.

Obrázek C.1: Úvodní stránka s aktuálními audity uživatele

		<b>HSE - HODNOCENÍ 5S A HSE NA PRACOVÍŠTI</b>						<b>NAPLÁNOVÁN</b> 17.03.2017		
								<b>PROVEDEN</b>		
								<b>UZAVŘEN</b>		
								<b>EDITOVÁN</b> -----		
<b>AUDITOR</b> VALENA		<b>BUŇKA</b> A25	<b>MISTR</b> KAMIS	<b>PŘEDÁK</b> <input type="text"/>	<b>HODNOCENÝ</b> <input type="text"/>	<b>%</b>	<b>STAV</b> NAPLÁNOVANÝ(0%)			
<b>KATEGORIE</b>	<b>#</b>	<b>OTÁZKA</b>					<b>OK</b>	<b>NOK</b>	<b>N/A</b>	
Uspořádanost pracoviště	1	Je na pracovišti určen vhodný způsob ukládání náradí (vc. přípravku na přeseržení)?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Odstranění nepotřebného	2	Je z pracoviště odstraněna neaktuální dokumentace?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Odstranění nepotřebného	3	Vyskytují se na pracovišti pouze věci související s výrobním procesem?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Uspořádanost pracoviště	4	Je na pracovišti určen vhodný způsob ukládání meridel?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Uspořádanost pracoviště	5	Je na pracovišti definované místo a způsob skladování materiálu?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Uspořádanost pracoviště	6	Je na pracovišti určen vhodný způsob ukládání dokumentace?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Čistota	7	Jsou podlahy na daném pracovišti čisté?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Čistota	8	Je stroj udržován v čistotě?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								
Čistota	9	Jsou meridla udržována v čistotě?					<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	
		KOMENTÁŘ: <input type="text"/>								

Obrázek C.2: Formulář HSE auditu.

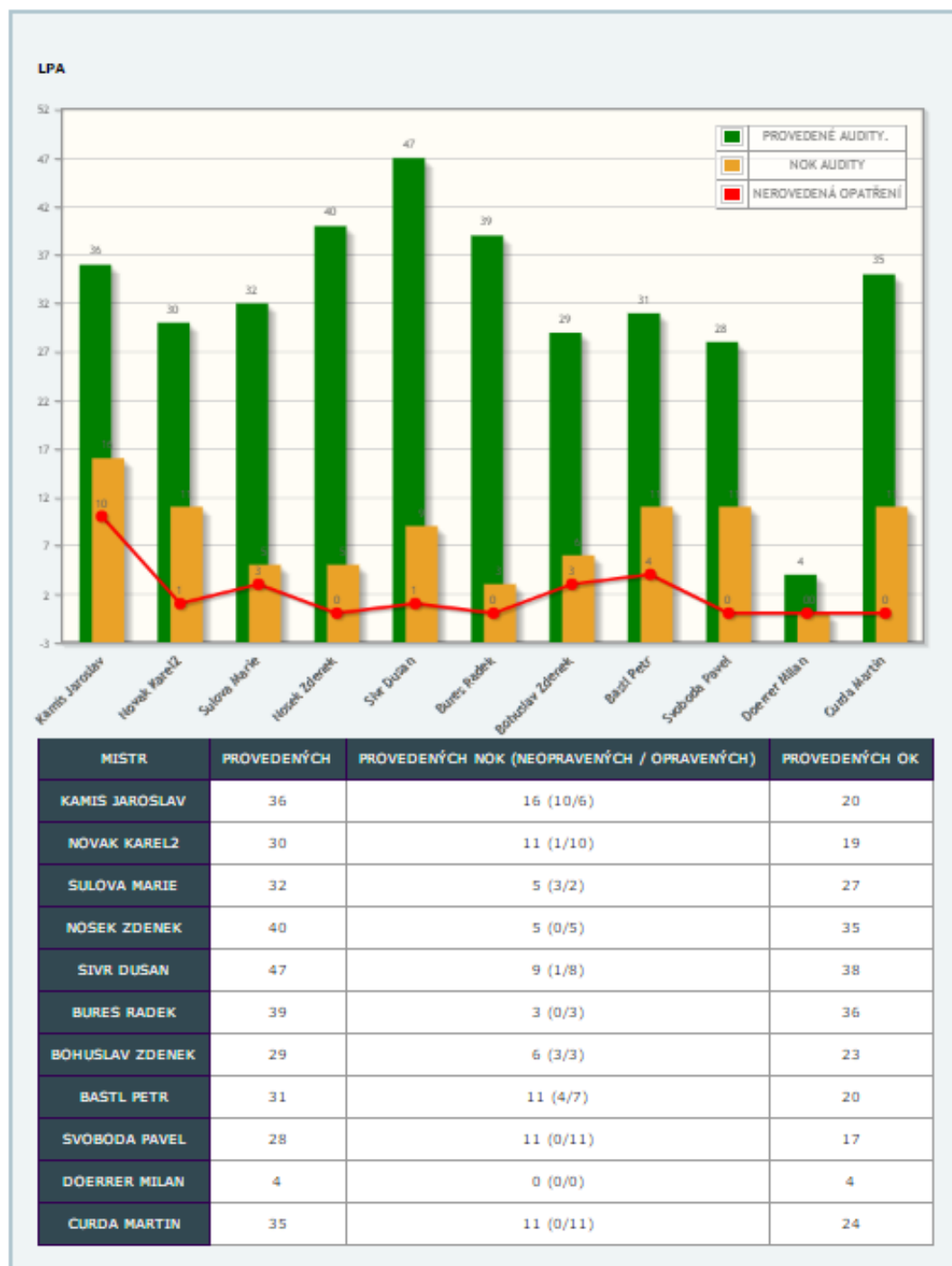
ADMINISTRACE	PLÁNOVÁNÍ	AUDITY	STATISTIKY a GRAFY	Roman Valera(odhlásit)
Aktuální audity	Moje audity	Zobrazit všechny audity		

## Seznam všech auditů

TYP AUDITU	LEVEL	BUŇKA	MISTR	AUDITOR	OD	DO						
..vyberte.. ▼	..vyberte.. ▼	..vyberte.. ▼	..vyberte.. ▼	..vyberte.. ▼	15.03.2017	31.05.2017						
<input type="checkbox"/> NAPLÁNOVANÝ <input type="checkbox"/> VYŘEŠENÝ OK <input type="checkbox"/> ŘEŠÍ SE <input type="checkbox"/> VYŘEŠENÝ NOK												
VYMAZAT FILTR											Filtrovat	
#	TYP	BUŇKA	MISTR	AUDITOR	VÝSLEDEK	STAV %	ŘEŠITEL	NAPLÁNOVÁN	PROVEDEN	VYŘEŠEN	UZAVŘEN	DETAIL
1	LPA1	C57	BURESR	SYROVP	100%	VYŘEŠENÝ OK		15.03.2017	15.03.2017	15.03.2017	ANO	<a href="#">DETAIL</a>
2	LPA1	C73	BURESR	SYROVP	93%	VYŘEŠENÝ NOK	VENUTA PETR	15.03.2017	15.03.2017	31.03.2017	ANO	<a href="#">DETAIL</a>
3	LPA1	H73	BURESR	SYROVP	100%	VYŘEŠENÝ OK		15.03.2017	15.03.2017	15.03.2017	ANO	<a href="#">DETAIL</a>
4	LPA1	M02	CURDAM	SYROVP	100%	VYŘEŠENÝ OK		15.03.2017	15.03.2017	15.03.2017	ANO	<a href="#">DETAIL</a>
5	LPA1	M74	CURDAM	SYROVP	100%	VYŘEŠENÝ OK		15.03.2017	15.03.2017	15.03.2017	ANO	<a href="#">DETAIL</a>
6	LPA1	C71	NOVAKK2	SYROVP	86%	VYŘEŠENÝ NOK	KRIZEK PAVEL	16.03.2017	16.03.2017	20.03.2017	ANO	<a href="#">DETAIL</a>
7	LPA1	C36	NOVAKK2	SYROVP	100%	VYŘEŠENÝ OK		16.03.2017	16.03.2017	16.03.2017	ANO	<a href="#">DETAIL</a>
8	LPA1	C47	NOVAKK2	SYROVP	100%	VYŘEŠENÝ OK		16.03.2017	16.03.2017	16.03.2017	ANO	<a href="#">DETAIL</a>
9	LPA1	A27	KAMISJ	SYROVP	64%	ŘEŠÍ SE(64%)	FUCIK MARTIN	16.03.2017	16.03.2017		NE	<a href="#">DETAIL</a>
10	LPA1	A04	KAMISJ	SYROVP	64%	ŘEŠÍ SE(64%)	FUCIK MARTIN	16.03.2017	16.03.2017		NE	<a href="#">DETAIL</a>
11	LPA1	A21	KAMISJ	SYROVP	100%	VYŘEŠENÝ OK		16.03.2017	16.03.2017	16.03.2017	ANO	<a href="#">DETAIL</a>
12	LPA1	P01	DOERRM	SYROVP	100%	VYŘEŠENÝ OK		16.03.2017	15.03.2017	15.03.2017	ANO	<a href="#">DETAIL</a>
13	LPA1	A13	KAMISJ			NAPLÁNOVANÝ		17.03.2017			NE	<a href="#">DETAIL</a>
14	LPA1	M68	NOSEKZ	SYROVP	100%	VYŘEŠENÝ OK		17.03.2017	17.03.2017	17.03.2017	ANO	<a href="#">DETAIL</a>
15	LPA1	M63	NOSEKZ	SYROVP	100%	VYŘEŠENÝ OK		17.03.2017	17.03.2017	17.03.2017	ANO	<a href="#">DETAIL</a>
16	LPA1	M46	NOSEKZ	SYROVP	100%	VYŘEŠENÝ OK		17.03.2017	17.03.2017	17.03.2017	ANO	<a href="#">DETAIL</a>
17	LPA1	H45	SIVRD	SYROVP	93%	VYŘEŠENÝ NOK		20.03.2017	20.03.2017	21.03.2017	ANO	<a href="#">DETAIL</a>
18	LPA1	H35	SIVRD	SYROVP	100%	VYŘEŠENÝ OK		20.03.2017	20.03.2017	20.03.2017	ANO	<a href="#">DETAIL</a>
19	LPA1	H75	SULOVM	SYROVP	93%	VYŘEŠENÝ NOK		20.03.2017	20.03.2017	10.05.2017	ANO	<a href="#">DETAIL</a>
20	LPA1	H74	SULOVM	HAVLIM	93%	ŘEŠÍ SE(93%)		20.03.2017	20.03.2017		ANO	<a href="#">DETAIL</a>
21	LPA1	C32	NOVAKK2	SYROVP	100%	VYŘEŠENÝ OK		20.03.2017	23.03.2017	23.03.2017	ANO	<a href="#">DETAIL</a>
22	LPA2	H29	NOSEKZ	FISEROVM	100%	VYŘEŠENÝ OK		20.03.2017	22.03.2017	22.03.2017	ANO	<a href="#">DETAIL</a>
23	HSE	A18	KAMISJ	KARPIMI	100%	VYŘEŠENÝ OK		20.03.2017	21.03.2017	21.03.2017	ANO	<a href="#">DETAIL</a>
24	LPA1	C70	NOVAKK2	SYROVP	93%	VYŘEŠENÝ NOK		21.03.2017	23.03.2017	04.04.2017	ANO	<a href="#">DETAIL</a>
25	LPA1	H69	BASTLP	SYROVP	93%	ŘEŠÍ SE(93%)	LEITKEP MILOS	21.03.2017	21.03.2017		NE	<a href="#">DETAIL</a>
26	LPA1	H69	BASTLP	SYROVP	100%	VYŘEŠENÝ OK		21.03.2017	21.03.2017	21.03.2017	ANO	<a href="#">DETAIL</a>
27	LPA1	H69	BASTLP	SYROVP	100%	VYŘEŠENÝ OK		21.03.2017	21.03.2017	21.03.2017	ANO	<a href="#">DETAIL</a>
28	LPA1	H54	BURESR	SYROVP	100%	VYŘEŠENÝ OK		21.03.2017	21.03.2017	21.03.2017	ANO	<a href="#">DETAIL</a>
29	LPA1	C281	BURESR	SYROVP	100%	VYŘEŠENÝ OK		21.03.2017	21.03.2017	21.03.2017	ANO	<a href="#">DETAIL</a>
30	LPA1	M73	BURESR	SYROVP	100%	VYŘEŠENÝ OK		22.03.2017	22.03.2017	22.03.2017	ANO	<a href="#">DETAIL</a>
31	LPA1	M39	BOHUSZ	SYROVP	93%	VYŘEŠENÝ NOK		22.03.2017	22.03.2017	10.04.2017	ANO	<a href="#">DETAIL</a>
32	LPA1	A36	KAMISJ	SYROVP	93%	ŘEŠÍ SE(93%)	FUCIK MARTIN	22.03.2017	22.03.2017		NE	<a href="#">DETAIL</a>
33	LPA1	A18	KAMISJ	SYROVP	100%	VYŘEŠENÝ OK		22.03.2017	22.03.2017	22.03.2017	ANO	<a href="#">DETAIL</a>
34	LPA1	A33	KAMISJ	SYROVP	100%	VYŘEŠENÝ OK		22.03.2017	22.03.2017	22.03.2017	ANO	<a href="#">DETAIL</a>
35	LPA1	M28	SIVRD	SYROVP	100%	VYŘEŠENÝ OK		22.03.2017	20.03.2017	20.03.2017	ANO	<a href="#">DETAIL</a>
36	LPA1	M81	BURESR	SYROVP	100%	VYŘEŠENÝ OK		23.03.2017	21.03.2017	21.03.2017	ANO	<a href="#">DETAIL</a>
37	LPA1	C40	NOVAKK2	SYROVP	100%	VYŘEŠENÝ OK		23.03.2017	23.03.2017	23.03.2017	ANO	<a href="#">DETAIL</a>

Obrázek C.3: Seznam auditů – zapnuté filtrování

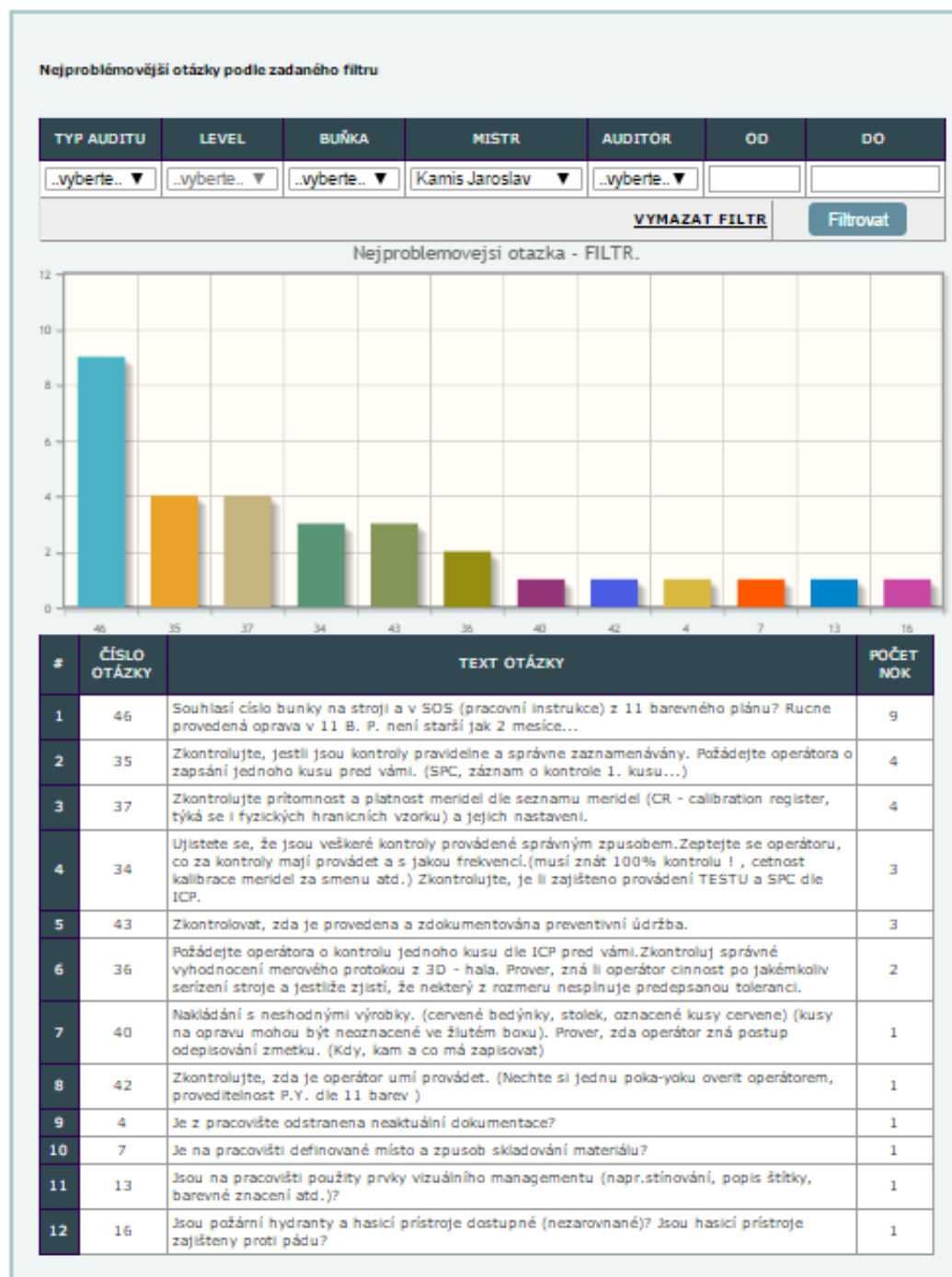
### Provedené / z toho NOK / z toho neopravených



Obrázek C.4: Provedené, problémové a neřešené audity v závislosti na mistrovi



## Nejproblémovější otázky



Obrázek C.5: Nejproblémovější otázky - zapnuté filtrování